

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

«На правах рукопису»  
УДК \_\_\_\_\_

«До захисту допущено»

Завідувач кафедри  
\_\_\_\_\_ І.Р. Пархомей  
(підпис)

“ \_\_\_\_ ” \_\_\_\_\_ 2019 р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

зі спеціальності 121 «Інженерія програмного забезпечення»

на тему: Інтелектуальна система розпізнавання візуальних ключів для мобільних додатків

Виконав: студент другого курсу, групи ІТ-84мп  
(шифр групи)

\_\_\_\_\_ Олексій Андрій Васильович

(прізвище, ім'я, по батькові)

\_\_\_\_\_ (підпис)

Науковий керівник доцент, к.т.н., доцент Корнага Я.І.  
(посада, науковий ступінь, вчене звання, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Консультант \_\_\_\_\_  
(назва розділу) (науковий ступінь, вчене звання, , прізвище, ініціали)

\_\_\_\_\_ (підпис)

Рецензент \_\_\_\_\_  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

\_\_\_\_\_ (підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент \_\_\_\_\_  
(підпис)

Київ – 2019 року

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»**

Факультет інформатики та обчислювальної техніки

Кафедра технічної кібернетики

Рівень вищої освіти – другий (магістерський)

Спеціальність 121 «Інженерія програмного забезпечення»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ І.Р. Пархомей

(підпис)

«\_\_\_» \_\_\_\_\_ 2019 р.

**ЗАВДАННЯ  
на магістерську дисертацію студенту**

Олексію Андрію Васильовичу

(прізвище, ім'я, по батькові)

1. Тема дисертації «Інтелектуальна система розпізнавання візуальних ключів для мобільних додатків», \_\_\_\_\_

науковий керівник дисертації доцент, к.т.н., доцент Корнага Я. І., \_\_\_\_\_  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «\_\_\_» \_\_\_\_\_ 2019 р. № \_\_\_\_\_

2. Термін подання студентом дисертації \_\_\_\_\_

3. Об'єкт дослідження – розпізнавання візуальних ключів

4. Предмет дослідження – методи розпізнавання зображень

5. Перелік завдань, які потрібно розробити – аналіз проблеми та існуючих рішень; аналіз і реалізація алгоритму; розробка алгоритму; дослідження ефективності розробленого алгоритму.

6. Орієнтовний перелік ілюстративного матеріалу – шість плакатів

7. Орієнтовний перелік публікацій – дві публікації

## 8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

9. Дата видачі завдання \_\_\_\_\_

## Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Аналіз предметної області	13.09.2019 р.	
2	Постановка задачі	15.09.2019 р.	
3	Аналіз інформаційного забезпечення	20.09.2019 р.	
5	Аналіз алгоритмічного забезпечення	25.09.2019 р.	
6	Розробка алгоритмічного забезпечення	15.10.2019 р.	
7	Розробка програмного забезпечення	01.11.2019 р.	
8	Маркетинговий аналіз стартап-проекту	10.11.2019 р.	
9	Висновки	15.11.2019 р.	

Студент

\_\_\_\_\_ (підпис)

А. В. Олексій  
(ініціали, прізвище)

Науковий керівник дисертації

\_\_\_\_\_ (підпис)

Я. І. Корнага  
(ініціали, прізвище)

## АНОТАЦІЯ

У роботі розглянуто проблему інтелектуального розпізнавання візуальних ключів.

У розділі аналізу предметної області та постановки задачі розглянуто існуючі підходи для інтелектуального розпізнавання зображень та розпізнавання об'єктів на цих зображеннях. В процесі було доведено доцільність використання згортоквих нейронних мереж для знаходження об'єктів.

Розділ аналізу інформаційного забезпечення присвячений опису існуючих методів розпізнавання та класифікації зображень, а також опису існуючих інструментів для реалізації відповідного алгоритму.

У розділі імплементації алгоритму розглянуто різні інструменти для розробки та налаштування нейронної мережі, прикладного рівня та інфраструктурного рівня системи. Також визначено архітектуру інтерфейсу додатку та розроблені вимоги до системи.

У розділі маркетингового аналізу стартап-проекту проаналізовано поточну ситуацію на ринку, розроблено стратегії та маркетингові плани для впровадження даного рішення.

Ключові слова: згорткова нейронна мережа, розпізнавання візуальних ключів.

Розмір пояснювальної записки – 85 аркушів, містить 19 ілюстрацій, 23 таблиці, 2 додатки (один з яких це графічний матеріал, що містить 4 плакати).

## ABSTRACT

The paper deals with the problem of intellectual recognition of visual keys.

The subject area analysis and formulation section discusses existing approaches for intelligent image recognition and object recognition in these images. The process proved the feasibility of using convolutional neural networks to find objects.

The information analysis section deals with the description of existing methods of image recognition and classification, as well as the description of existing tools for implementing the corresponding algorithm.

The algorithm implementation section covers various tools for developing and configuring a neural network, application layer, and system infrastructure layer. It also defines the application interface architecture and system requirements.

The marketing analysis section of the startup project analyzes the current market situation, develops strategies and marketing plans to implement this solution.

Keywords: convolutional neural network, recognition of visual keys.

The size of the explanatory note is 85 sheets, contains 19 illustrations, 23 tables, 2 attachments (one of them is graphic material containing 4 posters).

**Пояснювальна записка  
до магістерської дисертації**

на тему: *Інтелектуальна система розпізнавання візуальних ключів  
для мобільних додатків*

Київ – 2019 року

## ЗМІСТ

ВСТУП.....	9
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ .....	11
1.1 Об'єкт та предмет дослідження .....	11
1.2 Огляд існуючих рішень.....	12
1.2.1 Машинне навчання та нейронні мережі.....	12
1.2.2 Комп'ютерний зір та згорткові нейронні мережі.....	19
1.3 Розпізнавання об'єктів за допомогою CNN.....	26
1.3.1 R-CNN та Fast R-CNN .....	26
1.3.2 Генерування та використання пропозицій регіону .....	29
1.3.3 Просунуте згорткове знаходження об'єктів .....	33
1.4 Постановка задачі .....	35
Висновки по розділу .....	36
РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ .....	37
2.1 Критерій вибору.....	37
2.2 Знаходження об'єктів .....	39
2.3 Генерація регіонів.....	41
2.4 Оцінка об'єктів в контексті зображення .....	43
2.5 Датасети .....	46
2.6 Показники оцінки .....	49
Висновки по розділу .....	51
РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ АЛГОРИТМУ .....	52
3.1 MatConvNet та Fast R-CNN .....	52

3.2 Вибіркові поля пошуку та Edge Boxes .....	53
3.3 Геометричні висновки .....	54
3.5 Оцінка алгоритму .....	55
3.6 API дизайн та взаємодія з мобільним додатком .....	57
3.7 Вимоги до платформи.....	60
3.8 Вимоги до платформи розміщення API.....	61
Висновки по розділу .....	61
РОЗДІЛ 4. ОЦІНКА АЛГОРИТМУ.....	62
4.1 Fast R-CNN з різними регіонами .....	62
4.2 Оцінка коректності роботи.....	64
4.4 Геометричні висновки .....	67
Висновки по розділу .....	71
РОЗДІЛ 5. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ .....	72
5.1 Опис ідеї проекту .....	72
5.2 Технологічний аудит ідеї проекту.....	73
5.3 Аналіз ринкових можливостей запуску стартап-проекту.....	73
5.4 Розроблення ринкової стратегії проекту .....	80
5.5 Розроблення маркетингової програми стартап-проекту.....	82
Висновки по розділу .....	84
ВИСНОВКИ.....	85
ПЕРЕЛІК ПОСИЛАНЬ .....	86
ДОДАТКИ.....	89



## ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

CNN – згорткові нейронні мережі;

FCN – повністю з’єднана згорткова нейронна мережа;

FPS – кадри в секунду;

GPU – графічний процесор;

RPN – мережа запропонування регіонів;

Cache – проміжний буфер з швидким доступом, що містить інформацію, яка може бути запрошена з найбільшою ймовірністю;

БД – база даних;

API (application programming interface) – програмний інтерфейс;

ІТ – інформаційні технології.

## ВСТУП

У світі постійно зростає кількість даних про зображення та сам темп цього зросту. У 2019 році понад 1,8 мільярда зображень щодня завантажували найпопулярніші платформи, такі як Instagram та Facebook. Щодо відеоданих – згідно інформації компанії Google, на сервіс YouTube кожного дня завантажується більше 86 тисяч годин відео. В світі є безліч девайсів різних типів, що можуть робити фото та відео зйомку. Персональні смартфони та інші девайси мають змогу створювати знімки з великим розширенням, фототехніка доступна як ніколи. Також цілком доступним стає відеоустаткування з функцією автоматичного захоплення зображень. Автомобілі устатковуються камерами, що дають змогу ефективно слідкувати за трафіком та попереджувати нещасні випадки.

Щоб ефективно керувати всіма цими даними, нам потрібно мати деяке уявлення про їх зміст, а саме про об'єкти та деталі, що присутні на зображенні. Автоматизована обробка вмісту зображення корисна для широкого різноманіття завдань, пов'язаних із зображеннями. Для комп'ютерних систем це означає схрещування так званого семантичного розриву між інформацією про пікселі, що зберігається в файлі та розумінням інформаційного наповнення зображення.

Автоматична обробка дає можливість ідентифікувати об'єкти, що містяться у файлах зображень, опрацювати та описати деталі та параметри цих об'єктів. Це називається виявленням об'єктів і є однією з основних проблем комп'ютерного зору. Автоматичним системам та роботам є критично важливою можливість оцінювати візуальну сцену навколо себе та мати змогу аналізувати простір навколо себе, опрацьовувати інформацію.

Як ми продемонструємо, в даний час згорткові нейронні мережі є сучасним рішенням для виявлення об'єктів. В цій дисертації буде розкрито роботу

інтелектуальної системи розпізнавання візуальних ключів для мобільних додатків, а саме виявлення об'єктів на зображеннях за допомогою згорткових нейронних мереж.

Також буде проведено аналіз роботи нейронних систем, виявлено, як можна навчити цю мережу розпізнавати об'єкти, потрібні для процесу верифікації, як можна покращити роботу цього алгоритму за допомогою самонавчання нейронної мережі, питання побудови інфраструктури даної системи та які можуть бути помилки роботи нейронної мережі у разі помилок. Загалом дисертація має розкрити питання призначення згорткової нейронної мережі та доцільність її використання в поставленій задачі.

## РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ І ПОСТАНОВКА ЗАДАЧІ

### 1.1 Об'єкт та предмет дослідження

Розпізнавання образів є однією з найфундаментальніших проблем теорії інтелектуальних систем. З іншого боку, задача розпізнавання образів має величезне практичне значення. Замість терміну "розпізнавання" часто використовується термін "класифікація". Ці два терміни у багатьох випадках розглядаються як синоніми, але не є повністю взаємозамінюваними. Кожний з цих термінів має свої сфери застосування, і інтерпретація обох термінів часто залежить від специфіки конкретної задачі.

Задача ідентифікації, яка полягає у тому, щоб вирізнити певний конкретний об'єкт серед йому подібних (наприклад, впізнати серед інших людей свою дружину) та віднесення об'єкта до того чи іншого класу. Це може бути, наприклад, задача розпізнавання літер або прийняття рішення про наявність дефекту у деякій технічній деталі. Віднесення об'єкта до певного класу відображає найтипівішу проблему класифікації, і, коли говорять про розпізнавання образів, найчастіше мають на увазі саме цю проблему. Саме вона розглядається тут у першу чергу.

Кластерний аналіз, який полягає в розділенні заданого набору об'єктів на класи, групи об'єктів, схожі між собою за тим чи іншим критерієм. Цю задачу часто називають класифікацією без учителя, оскільки, на відміну від задачі 2, класи апіорно не задані.

Проблеми розпізнавання легко вирішуються людьми, причому, робиться це, як правило, підсвідомо. Однак, спроби побудувати штучні системи розпізнавання не настільки переконливі. Основна проблема полягає у тому, що у більшості випадків неможливо адекватно визначити ознаки, на основі яких слід здійснювати розпізнавання. Для задач, для яких такі ознаки вдається виділити, штучні системи розпізнавання набули значного поширення і широко використовуються.

В випадку даної роботи основною задачею є ідентифікація об'єкта візуального ключа на зображенні, тобто вирішення завдання класифікації. Варто сказати, що об'єктом даної роботи є розпізнавання візуальних ключів в вигляді конкретних об'єктів, а предметом – конкретні методи розпізнавання зображень, а саме згорткові нейронні мережі, про які буде йти мова нижче, в розділі 2.2.

## 1.2 Огляд існуючих рішень

У цій главі ми надаємо теоретичну основу, необхідну для розуміння методів, обговорених у наступному розділі. Спочатку ми обговорюємо відповідні деталі машинного навчання, нейронних мереж та комп'ютерного зору. Нарешті, ми пояснимо, як ці дисципліни поєднуються у згорткових нейронних мережах.

### 1.2.1 Машинне навчання та нейронні мережі

Алгоритми навчання широко використовуються в програмах комп'ютерного зору. Перш ніж розглядати завдання, пов'язані із зображенням, ми коротко ознайомимося з основами машинного навчання.

Машинне навчання стало корисним інструментом для моделювання проблем, які інакше складно точно сформулювати. Класичні комп'ютерні програми явно програмуються вручну для виконання завдання. При машинному навчанні деяка частина людського внеску замінюється алгоритмом навчання. Із збільшенням доступності обчислювальної спроможності та даних зростає, з часом машинне навчання стає все більш практичним, до того моменту, що воно стає майже всюдишним.

Алгоритм навчання показано через кілька прикладів, які були помічені або визначені людиною. Наприклад, у проблемі виявлення об'єкта ми використовуємо навчальні зображення, де люди позначили місця та класи на відповідні об'єкти. Дізнавшись на прикладах, алгоритм вміє передбачити примітки або мітки раніше небачених даних.

Типовим способом використання машинного навчання є наглядове навчання. Алгоритм навчання показано кілька прикладів, які були позначені або позначені людиною. Наприклад, у проблемі виявлення об'єктів ми використовуємо навчальні зображення, де люди позначили місця та класи відповідних об'єктів. Дізнавшись з прикладів, алгоритм здатний передбачити анотації або мітки раніше небачених даних.

Класифікація та регресія це найважливіші види завдань. У класифікації алгоритм намагається передбачити правильний клас нового фрагмента даних на основі даних тренувань. При регресії замість дискретних класів алгоритм намагається передбачити безперервний вихід. Під час навчання без нагляду алгоритм намагається засвоїти корисні властивості даних без того, щоб вчитель людини повідомив, яким повинен бути правильний вихід.

Класичним прикладом неуспішного навчання є кластеризація. Останнім часом, особливо з появою технологій глибокого навчання, непідконтрольна попередня обробка стала популярним інструментом у контрольованих навчальних завданнях для виявлення корисних уявлень даних.

Якась попередня обробка майже завжди потрібна. Попередня обробка даних у новий, простіший змінний простір називається вилученням функцій. Часто використовувати повномірні дані тренінгу недоцільно або неможливо. Швидше за все, детектори запрограмовані для отримання цікавих особливостей з даних, і ці функції використовуються як вхід до алгоритму машинного навчання.

У минулому детектори функцій часто виготовлялися вручну. Проблема такого підходу полягає в тому, що ми не завжди заздалегідь знаємо, які особливості цікаві. Тенденція машинного навчання полягала і в вивченні функцій детекторів, що дозволяє використовувати повні дані.

Оскільки дані тренінгу не можуть включати всі можливі екземпляри вхідних даних, алгоритм навчання повинен бути у змозі узагальнити для обробки невидимих точок даних. Занадто проста оцінка моделі може не змогти зафіксувати

важливі аспекти справжньої моделі. З іншого боку, занадто складні методи можуть переоцінити, моделюючи неважливі деталі та шум, що також призводить до поганого узагальнення.

Як правило, перевитрата трапляється тоді, коли використовується складний метод у поєднанні з занадто низькою кількістю навчальних даних. Переобладнана модель вчиться моделювати відомі приклади, але не розуміє, що їх пов'язує. Продуктивність алгоритму можна оцінити за якістю та кількістю помилок. Функція втрат, наприклад середня квадратна помилка, використовується для присвоєння вартості помилкам. Завданням на етапі тренінгу є мінімізація цієї втрати.

Нейронні мережі – популярний тип моделі машинного навчання. Особливий випадок нейронної мережі називається згортковою нейронною мережею (CNN). Перш ніж обговорити CNN пояснимо, що таке штучні нейронні мережі.

Штучні нейронні мережі це обчислювальні системи, натхнені біологічними нейронними мережами, що складають мозок тварин. Такі системи навчаються задач (поступально покращують свою продуктивність на них), розглядаючи приклади, загалом без спеціального програмування під задачу. Наприклад, у розпізнаванні зображень вони можуть навчатися ідентифікувати зображення, які містять котів, аналізуючи приклади зображень, мічені[en] як «кіт» і «не кіт», і використовуючи результати для ідентифікування котів в інших зображеннях. Вони роблять це без жодного апіорного знання про котів, наприклад, що вони мають хутро, хвости, вуса та котоподібні пискі.

Нейронні мережі спочатку називали штучними нейронними мережами, оскільки вони були розроблені для імітації нервової функції мозку людини. Незважаючи на те, що натхнення з біології є очевидним, було б оманливим перебільшувати зв'язок між штучними нейронами та біологічними нейронами або нейронаукою. Мозок людини містить приблизно 100 мільярдів нейронів, що працюють паралельно. Штучні нейрони – це математичні функції, реалізовані на

більш-менш послідовних комп'ютерах. Дослідження в нейронні мережах в основному керуються розробками в галузі інженерії та математики, а не біології.

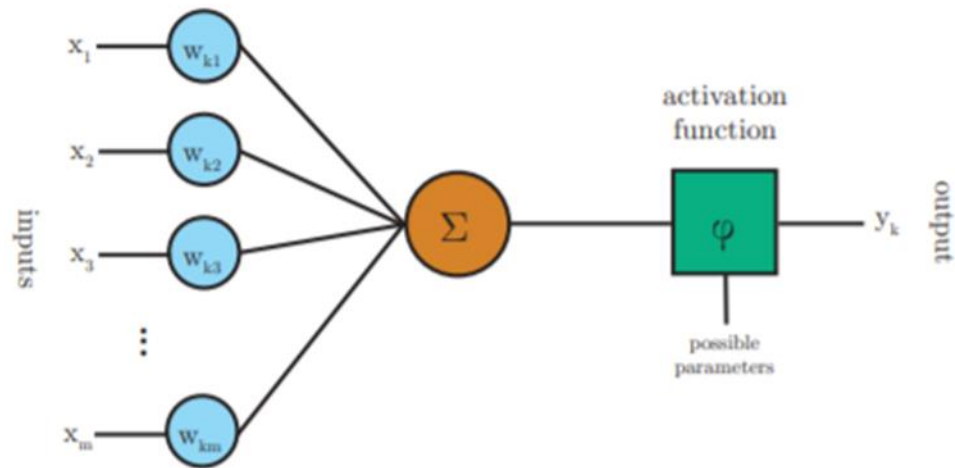


Рисунок 1.1. Схема нейрона

Штучний нейрон на основі моделі МакКаллоха-Піттса показаний на рис. 1.1. Нейрон  $k$  приймає  $t$  вхідних параметрів  $x$ . Нейрон також має  $t$  вагові параметри  $W_{kt}$ . Вагові параметри часто включають термін зміщення, який має відповідні манекени з фіксованим значенням 1. Введення та ваги лінійно поєднуються та підсумовуються.

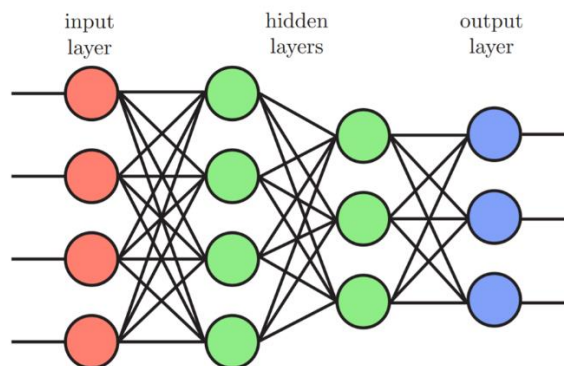


Рисунок 1.2. Схема повністю пов'язаної багатошарової нейронної мережі



Нейронна мережа це поєднання штучних нейронів. Нейрони, як правило, групуються в шари. У повністю пов'язаній багатошаровій мережі з подачею вперед, показаною на рис. 1.2, кожен вихід шару нейронів подається як вхід до кожного нейрона наступного шару. Таким чином, деякі шари обробляють вихідні вхідні дані, тоді як деякі обробляють дані, отримані від інших нейронів. Кожен нейрон має кількість ваг, рівних кількості нейронів у попередньому шарі.

Багатошарова мережа, як правило, включає три типи шарів: вхідний шар, один або більше прихованих шарів та вихідний шар. Вхідний рівень зазвичай просто передає дані, не змінюючи їх. Більша частина обчислень відбувається в прихованих шарах. Вихідний рівень перетворює приховані активації шару у вихід, наприклад, у класифікацію. Багатошарова мережа передачі подачі щонайменше з одним прихованим шаром може функціонувати як універсальний апроксиматор, тобто може бути побудована для обчислення практично будь-якої функції [20]. У цій роботі ми здебільшого обговоримо повністю пов'язані мережі та згорткові мережі.

Згорткові мережі використовують спільний доступ до параметрів і мають обмежене з'єднання порівняно з повністю підключеними мережами [21]. Інші типи мереж, наприклад, періодичні мережі, виходять за межі цієї роботи.

Нейронна мережа тренується шляхом вибору ваги всіх нейронів, щоб мережа вчилася наближати цільові виходи з відомих входів.

Важко вирахувати ваги нейронів багатошарової мережі аналітично. Алгоритм зворотного розповсюдження забезпечує просте і ефективне рішення розв'язування ваг в ітераційному порядку. Класична версія використовує градієнтне спуск як метод оптимізації. Спуск градієнта може бути досить трудомістким і не гарантовано знайти глобальний мінімум помилок, але при правильній конфігурації (відомий в машинному навчанні як гіперпараметри) працює досить добре на практиці.

На першій фазі алгоритму вхідний вектор поширюється вперед через нейронну мережу. До цього ваги мережевих нейронів були ініціалізовані до деяких значень, наприклад малих випадкових значень. Отриманий вихід мережі порівнюється з потрібним виходом (який повинен бути відомий на навчальних прикладах) з використанням функції втрат. Потім обчислюється градієнт функції втрат. Цей градієнт також називається значенням помилки. Використовуючи середню квадратичну помилку як функцію втрати, значення помилки вихідного шару це просто різниця між поточним та бажаним виходом.

Потім значення похибки поширюються назад через мережу для обчислення значень помилок нейронів прихованого шару. Градієнти функції прихованих втрат нейрона можуть бути вирішені за допомогою ланцюгового правила похідних. Нарешті, ваги нейронів оновлюються шляхом обчислення градієнта ваг і віднімання частки градієнта від ваг. Це співвідношення називається швидкістю навчання. Коефіцієнт навчання може бути фіксованим або динамічним. Після оновлення ваг алгоритм продовжує виконувати фази знову з різним введенням, поки ваги не збігаються.

У наведеному вище описі ми описали онлайн-навчання, яке розраховує оновлення ваги після кожного нового введення. Онлайн-навчання може призвести до поведінки «зигзагування», коли оцінка єдиної точки даних градієнта постійно змінює напрямок і не наближається до мінімуму безпосередньо. Інший спосіб обчислення оновлень це повне пакетне навчання, де ми обчислюємо вагові оновлення для повного набору даних. Це досить обчислювально важко і має інші недоліки. Компромісна версія це міні-серійне навчання, де ми використовуємо лише частину навчального набору для кожного оновлення [14].

Функція активації  $\phi$  визначає кінцевий вихід кожного нейрона. Важливо правильно вибрати функцію для створення ефективної мережі.

Ранні дослідники виявили, що перцептрони та інші лінійні системи мали серйозні недоліки, не в змозі вирішити проблеми, які не були лінійно

відокремлені, наприклад, проблема XOR. Іноді лінійні системи можуть вирішувати подібні проблеми за допомогою ручно виготовлених детекторів функцій, але це не найвигідніше використання машинного навчання. Просте додавання шарів також не допомагає, оскільки мережа, що складається з лінійних нейронів, залишається лінійною незалежно від того, скільки шарів вона має.

Легким та ефективним способом створення нелінійної мережі є використання випрямлених лінійних одиниць (ReLU). Випрямлена лінійна функція генерує вихід, використовуючи функцію рампи.

Цей тип функцій легко обчислити та диференціювати (для зворотного розповсюдження). Функція не є диференційованою за нуля, але це не завадило її використанню на практиці. Останнім часом ReLus стали досить популярними, часто замінюючи функції сигмоїдної активації, які мають гладкі похідні, але страждають від проблем насичення градієнта та повільніших обчислень. Для проблем класифікації класів багатofункціональна функція активації softmax.

Сучасні нейронні мережі часто називають глибокими нейронними мережами. Незважаючи на те, що багат шарові нейронні мережі існують з 1980-х років, кілька причин заважали ефективному навчанню мереж з декількома прихованими шарами [19]. Однією з головних проблем є проблема розмірності [20].

Зі збільшенням кількості змінних кількість різних конфігурацій змінних експоненціально зростає. Зі збільшенням кількості конфігурацій кількість навчальних зразків повинна збільшуватися в рівній мірі. Збір навчального набору даних достатнього розміру забирає багато часу та дорого чи відверто неможливо. На щастя, дані в реальному світі не розподілені рівномірно і часто включають структуру, де цікава інформація лежить на низькомірному колекторі.

Гіпотеза багатоманітності передбачає, що більшість конфігурацій даних є недійсними або рідкісними. Ми можемо зменшити розмірність, навчившись представляти дані, використовуючи координати колектора. Інший спосіб

вдосконалення узагальнення це припускати локальну сталість. Це означає припускати, що функція, яку нейронна мережа вчиться наближати, не повинна сильно змінюватися в межах малого регіону. За останні десять років нейронні мережі зазнали ренесансу, головним чином через наявність більш потужних комп'ютерів та більших наборів даних.

На початку 2000-х років було виявлено, що нейронні мережі можуть ефективно навчатися за допомогою графічних блоків обробки. Графічні процесори є більш ефективними для виконання завдання, ніж традиційні процесори, і пропонують відносно дешеву альтернативу спеціалізованому апаратному забезпеченню.

Сьогодні дослідники зазвичай використовують високоякісні графічні картки для споживачів, такі як NVIDIA Tesla K40. Інші більш теоретичні прориви включають заміну помилок середнього квадрату на функції помилок на основі перехресної ентропії та заміну сигмоїдальних функцій активації на випрямлені лінійні одиниці. При глибокому навчанні виникає менша потреба в налаштованих вручну рішеннях машинного навчання, які використовувалися раніше. Наприклад, класична система виявлення візерунків включає в себе ручну налаштування фази виявлення функцій перед фазою машинного навчання.

Еквівалент глибокого навчання складається з єдиної нейронної мережі. Нижні шари нейронної мережі вчать розпізнавати основні ознаки, які потім подаються вперед до вищих шарів мережі.

### 1.2.2 Комп'ютерний зір та згорткові нейронні мережі

Далі ми плануємо обговорити комп'ютерне бачення в цілому та дослідити основний предмет цієї роботи, виявлення об'єктів, як підпроблеми комп'ютерного зору.

Комп'ютерне бачення займається вилученням змістовної інформації із вмісту цифрових зображень чи відео. Це відрізняється від простої обробки

зображень, яка передбачає маніпулювання візуальною інформацією на рівні пікселів.

Застосування комп'ютерного зору включає класифікацію зображень, візуальне виявлення, реконструкцію 3D-сцени з двовимірних зображень, пошук зображень, доповнену реальність, машинне бачення та автоматизацію руху.

Сьогодні машинне навчання є необхідною складовою багатьох алгоритмів комп'ютерного зору. Такі алгоритми можна описати як поєднання обробки зображень та машинного навчання.

Для ефективних рішень потрібні алгоритми, які можуть впоратися з величезною кількістю інформації, що міститься у візуальних зображеннях, і критично для багатьох застосувань можуть здійснювати обчислення в режимі реального часу.

Виявлення об'єктів є однією з класичних проблем комп'ютерного зору і часто описується як складне завдання. Багато в чому він схожий з іншими завданнями комп'ютерного зору, оскільки передбачає створення рішення, інваріантного деформації та зміни освітлення та точки зору. Що робить виявлення об'єктів виразною проблемою, це те, що воно включає як локалізацію, так і класифікацію областей зображення [20]. Розміщувальна частина не потрібна, наприклад, у всій класифікації зображень.

Щоб виявити об'єкт, нам потрібно мати уявлення про те, де може бути об'єкт і як зображення сегментується. Це створює тип куряче-яєчної проблеми, де для розпізнавання форми (та класу) предмета нам потрібно знати його розташування, а для розпізнавання місця розташування предмета, нам потрібно знати його форму. Деякі види, що відрізняються візуально, такі як одяг і обличчя людини, можуть бути частинами одного предмета, але це важко пізнати, не розпізнавши спочатку предмет. З іншого боку, деякі об'єкти виділяються лише трохи від фону, вимагаючи розділення перед розпізнаванням.

Візуальні особливості зображення низького рівня, такі як карта схильності, можуть бути використані як посібник для розміщення кандидатських об'єктів. Місце і розмір, як правило, визначаються за допомогою обмежувального поля, яке зберігається у вигляді кутових координат. Використовувати прямокутник простіше, ніж використовувати багатокутник довільної форми, і багато операцій, таких як згортка, виконуються на прямокутниках у будь-якому випадку. Потім зображення, що міститься в обмежувальному полі, класифікується за алгоритмом, який був підготовлений за допомогою машинного навчання.

Межі об'єкта можна додатково уточнювати ітераційно, після початкової здогадки.

Протягом 2000-х популярні рішення для виявлення об'єктів використовували дескриптори функцій, такі як масштабне інваріантне перетворення ознак (SIFT), розроблений Девідом Лоу в 1999 році, і гістограму орієнтованих градієнтів (HOG), популяризовану в 2005 році. У 2010-х відбувся зсув у напрямку використання звивистих нейронних мереж.

Перед широкомасштабним прийняттям CNN існували два конкуруючих рішення для створення обмежувальних коробок. У першому рішенні формується щільний набір регіональних пропозицій, а потім більшість із них відхиляються. Зазвичай це детектор розсувних вікон. У другому рішенні генерується набір обмежувальних коробок за допомогою методу пропозиції регіону, такого як селективний пошук. Поєднання розрізнених пропозицій регіону із звивистими нейронними мережами дало хороші результати та є популярною в даний час.

Проблема вирішення проблем із комп'ютерним зором за допомогою традиційних нейронних мереж полягає в тому, що навіть скромне зображення містить величезну кількість інформації.

Монохромне зображення 620x480 містить 297 600 пікселів. Якщо інтенсивність кожного пікселя цього зображення вводиться окремо до повністю

пов'язаної мережі, кожному нейрону потрібно 297 600 ваг. Повне HD-зображення 1920x1080 вимагатиме 2 073 600 ваг. Якщо зображення поліхромні, кількість ваг множитья на кількість кольорових каналів (як правило, три). Таким чином, ми можемо бачити, що загальна кількість вільних параметрів у мережі швидко стає надзвичайно великою, оскільки розмір зображення збільшується. Занадто великі моделі викликають переобладнання і повільну продуктивність.

Крім того, багато завдань на виявлення шаблонів вимагають. Неефективно тренувати нейрони окремо розпізнавати ту саму схему в лівому верхньому куті та в правому нижньому куті зображення. Повністю пов'язана нейронна мережа не може прийняти такого роду структуру до уваги.

Основна ідея CNN була натхненна концепцією в біології під назвою сприйнятливий поле. Рецептивні поля – особливість зорової кори тварин. Вони виконують роль детекторів, чутливих до певних видів подразника, наприклад, ребер. Вони знаходяться поперек зорового поля та перекриваються один з одним.

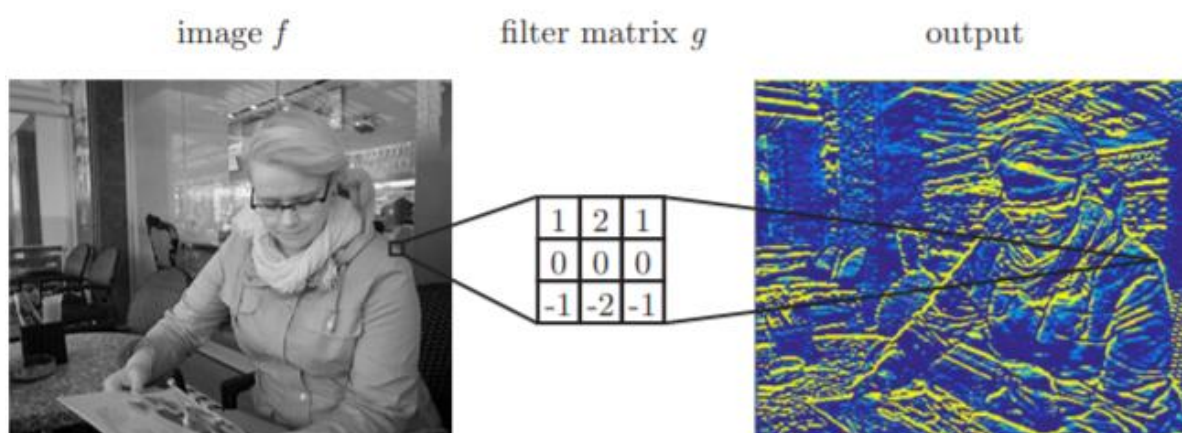


Рисунок 1.3. Виявлення горизонтальних кутів зображення

Цю біологічну функцію можна наблизити в комп'ютерах за допомогою операції згортання. При обробці зображень зображення можна фільтрувати за допомогою згортки для отримання різних видимих ефектів.

Набір згорткових фільтрів можна комбінувати для формування згорткового шару нейронної мережі. Матричні значення фільтрів розглядаються як параметри нейронів і навчаються за допомогою машинного навчання. Операція згортки замінює операцію множення звичайного шару нейронної мережі. Вихід шару зазвичай описується як об'єм. Висота та ширина гучності залежать від розмірів карти активації. Глибина гучності залежить від кількості фільтрів. Оскільки однакові фільтри використовуються для всіх частин зображення, кількість вільних параметрів різко зменшується порівняно з повністю пов'язаним нейронним шаром. Нейрони згорткового шару здебільшого мають однакові параметри і з'єднані лише з локальною областю входу. Обмін параметрами в результаті згортання забезпечує інваріантність перекладу. Альтернативний спосіб опису звивистого шару – це уявити повністю пов'язаний шар з нескінченно сильним попереднім розміщенням на його вагах. Це примушує нейрони ділити ваги в різних просторових місцях і мати нульову вагу поза полем сприйняття.

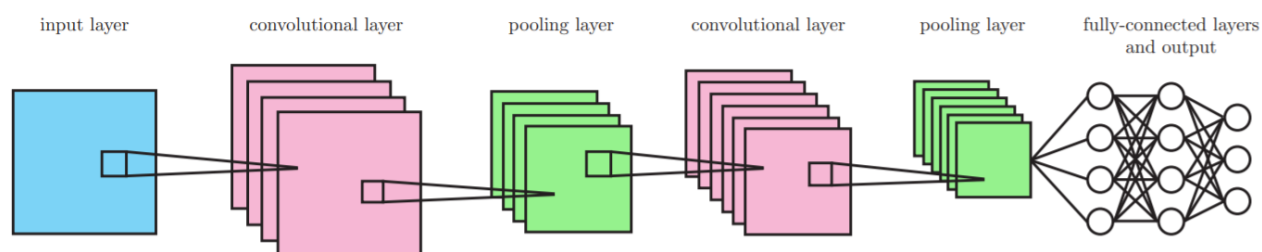


Рисунок 1.4. Приклад згорткової нейромережі

Послідовні згорткові шари (часто поєднуються з іншими типами шарів, наприклад, об'єднання, описане нижче) утворюють згорткову нейронну мережу (CNN). Приклад згорткової мережі показаний на рис. 1.4. Теоретично, шари, ближче до вводу, повинні навчитися розпізнавати особливості зображення на низькому рівні, такі як краї та кути, а шари, наближені до виводу, повинні навчитися поєднувати ці особливості для розпізнавання більш значущих форм. У



цій тезі нам цікаво вивчити, чи можуть згорткові мережі навчитися розпізнавати цілісні об'єкти.

Щоб зробити мережу більш керованою для класифікації, корисно зменшити розмір карти активації в глибокому кінці мережі. Взагалі, глибокі шари мережі вимагають менше інформації про точні просторові місця розташування ознак, але потребують більше матриць фільтру, щоб розпізнати кілька шаблонів високого рівня. За рахунок зменшення висоти та ширини об'єму даних ми можемо збільшити глибину обсягу даних та підтримувати час обчислень на розумному рівні. Існує два способи зменшення обсягу даних. Один із способів – включити об'єднаний шар після згорткового шару. Шар ефективно знижує проби карти активації. Об'єднання має додатковий ефект, що робить отриману мережу більш інваріантною трансляцією, змушуючи детектори бути менш точними. Однак об'єднання може зруйнувати інформацію про просторові зв'язки між підрозділами моделей. Типовим методом об'єднання є максимальне об'єднання. Макс-об'єднання просто виводить максимальне значення в прямокутному районі карти активації. Ще одним способом зменшення обсягу даних є коригування параметра кроку операції згортання. Параметр *stride* керує тим, чи обчислюється вихід згортки для мікрорайону, зосередженого на кожному пікселі вхідного зображення (*stride* 1), або для кожного *n*-го пікселя. Дослідження показали, що об'єднання шарів часто можна відкинути без втрати точності, використовуючи згорткові шари з більшим значенням кроку. Операція кроку еквівалентна використанню фіксованої сітки для об'єднання.

Звитий шар зазвичай включає нелінійну функцію активації, таку як випрямлена функція лінійної активації. Активації іноді описуються як окремий шар між згортковим шаром і шаром об'єднання. Деякі системи, такі як, також реалізують шар, який називається локальною нормалізацією реакції, який використовується як метод регуляризації. Локальна нормалізація реакції імітує функцію біологічних нейронів, звану бічним гальмуванням, внаслідок чого

збуджені нейрони знижують активність сусідніх нейронів. Однак інші методи регуляризації в даний час більш популярні, і вони обговорюються в наступному розділі.

Кінцеві приховані шари CNN, як правило, повністю пов'язані між собою шари. Повністю пов'язаний шар може охоплювати деякі цікаві відносини, згорнуті шари спільного використання параметрів не можуть. Однак для повного підключення шару потрібен досить невеликий об'єм даних, щоб бути практичним. Налаштування об'єднання та кроку можна використовувати для зменшення розміру обсягу даних, який досягає повністю пов'язаних шарів. Звита мережа, яка не включає повністю пов'язаних шарів, називається повністю згортковою мережею (FCN). Якщо мережа використовується для класифікації, вона зазвичай включає софтмакс вихідний шар. Активації самих верхніх шарів можуть також використовуватися безпосередньо для створення представленого зображення зображення. Це означає, що згорткова мережа використовується як великий детектор функцій.

Регуляризація стосується методів, що застосовуються для зменшення перевитрати, вводячи додаткові обмеження чи інформацію до системи машинного навчання. Класичний спосіб використання регуляризації в нейронних мережах – додавання штрафу до функції мета / втрата, яка карає певні типи ваг. Особливість спільного використання параметрів згорткових мереж – ще один приклад регуляризації.

Існує кілька методів регуляризації, характерних для глибоких нейронних мереж. Популярна методика під назвою випадання намагається зменшити коадаптацію нейронів. Це досягається випадковим випаданням нейронів під час тренування, що означає, що для кожної тренувальної вибірки або міні-партії використовується дещо інша нейронна мережа. Це призводить до того, що система не дуже сильно залежить від якогось одного нейрона або з'єднання і забезпечує ефективний, але обчислювально недорогий спосіб здійснення регуляризації. У

згорткових мережах випадання зазвичай використовується в кінцевих повністю з'єднаних шарах.

Перевиконання можна також зменшити за рахунок збільшення кількості даних про навчання. Коли неможливо придбати більше фактичних вибірок, розширення даних використовується для отримання більшої кількості вибірок із існуючих даних.

Для класифікації за допомогою згорткових мереж цього можна досягти, обчисливши перетворення вхідних зображень, які не змінюють сприйняті об'єкти класів, але створюють додаткову проблему для системи. Зображення можуть бути, наприклад, перевернуті, обернені або зразки з різними культурами та масштабами. Також до вхідних зображень може додаватися шум.

### 1.3 Розпізнавання об'єктів за допомогою CNN

У цій главі ми обговорюємо та порівнюємо різні методи виявлення об'єктів, які використовують звивисті нейронні мережі. Зокрема, ми розглянемо методи, що поєднують CNN з класифікацією пропозицій у регіонах. Далі ми обговорюємо, як формуються пропозиції регіонів, які також називаються регіонами, що цікавлять (RoI).

#### 1.3.1 R-CNN та Fast R-CNN

Обчислення прямого R-CNN має кілька етапів, показаних на рис. 1.5. По-перше, створюються цікаві регіони. RoI є категорією незалежних обмежувальних коробок, які мають велику ймовірність вмістити цікавий об'єкт. Для їх створення використовується окремий метод під назвою "Селективний пошук", але натомість можуть бути використані інші методи генерації регіонів. Вибірковий пошук, поряд з іншими методами формування пропозицій у регіонах, детальніше обговорюється в розділі 1.3.3.

Далі, згорнута мережа використовується для отримання функцій з кожної пропозиції регіону. Під-зображення, що міститься в обмежувальній коробці, викривляється, щоб відповідати вхідному розміру CNN, а потім подається в мережу. Після того, як мережа вилучила функції з входу, функції надходять на підтримку векторних машин (SVM), які забезпечують остаточну класифікацію.

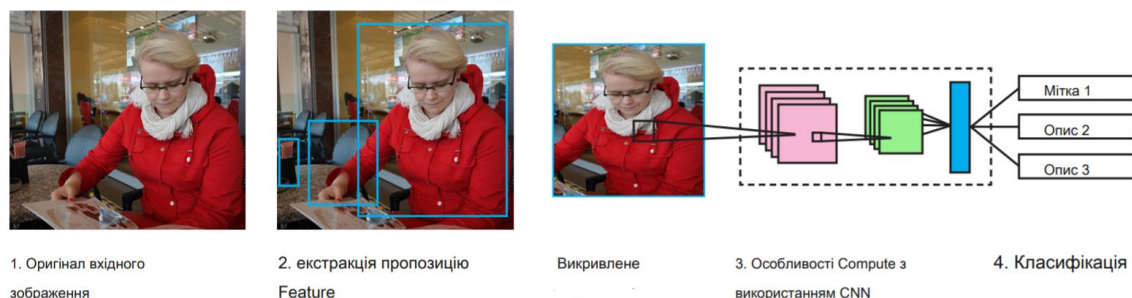


Рисунок 1.5 Етапи R-CNN обчислень

Метод тренується в кілька етапів, починаючи з згорткової мережі. Після того, як CNN пройшов навчання, SVM підходять до функцій CNN. Нарешті, метод формування генерації пропозицій регіону підготовлений.

R-CNN є важливим методом, оскільки він забезпечив перше практичне рішення для виявлення об'єктів за допомогою CNN. Будучи першим, він має багато недоліків, які були вдосконалені пізнішими методами. Є три основні проблеми R-CNN: По-перше, навчання складається з декількох етапів, як описано вище. По-друге, навчання дороге. Як для навчання SVM, так і для регіону, функції витягуються з кожної пропозиції регіону та зберігаються на диску. Для цього потрібні дні обчислень і сотні гігабайт місця для зберігання. По-третє, і, мабуть, найважливіше, виявлення об'єктів відбувається повільно, вимагаючи майже хвилини для кожного зображення, навіть на GPU. Це пояснюється тим, що обчислення прямої передачі CNN виконується окремо для кожної об'єктної пропозиції, навіть якщо пропозиції походять із одного зображення або перетинаються один з одним.

Загальна структура Fast R-CNN проілюстрована на рис. 1.6. Метод отримує в якості зображення зображення плюс області інтересу, обчислені з зображення. Як і в R-CNN, РІ генеруються за допомогою зовнішнього методу. Зображення обробляється за допомогою CNN, який включає декілька згорткових і максимуму шарів об'єднання. Згорнута карта особливостей, що формується після цих шарів, вводиться в шар об'єднання RoI. Це витягує функціональний вектор фіксованої довжини для кожного RoI з карти функцій. Потім вектори функцій вводяться до повністю з'єднаних шарів, які з'єднані з двома вихідними шарами: шару softmax, який створює оцінки ймовірності для класів об'єктів, і реального значення шару, який виводить обмежуючі поля, що координуються, обчислюються за допомогою регресії (тобто уточнення до початкових скриньки кандидатів).

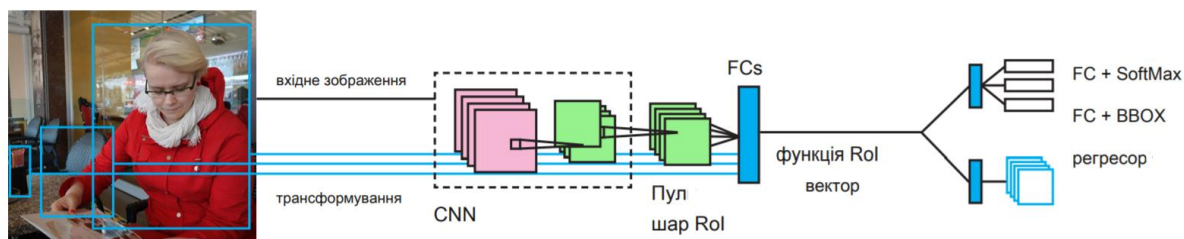


Рисунок 1.6. Етапи Fast R-CNN обчислень

На думку авторів, швидкий R-CNN забезпечує значно коротший час класифікації порівняно з звичайним R-CNN, забираючи менше ніж секунду на найсучасніший GPU. В основному це пов'язано з використанням однієї і тієї ж карти функцій для кожного IP. Із зменшенням часу виявлення загальний час обчислення починає значно залежати від ефективності методу формування пропозицій у регіоні. Крім того, коли існує багато RoI, час, витрачений на оцінку повністю пов'язаних шарів, може домінувати в часі оцінювання згорткових шарів. Час класифікації можна прискорити приблизно на 30%, якщо повністю з'єднані шари стискаються за допомогою усіченого сингулярного розкладання величини. Однак це призводить до незначного зниження точності.

Згідно з оригінальною публікацією [20], швидкий R-CNN є більш ефективним для тренувань, ніж R-CNN, з дев'ятикратним скороченням часу на тренування. Усю мережу (включаючи шар об'єднання RoI та повністю пов'язані шари) можна тренувати, використовуючи алгоритм зворотного поширення та стохастичний градієнтний спуск. Як правило, заздалегідь підготовлена мережа використовується як вихідний пункт, а потім тонко налаштована. Навчання проводиться в міні-партіях з  $N$  зображень.  $R / N$  RoI відбираються з кожного міні-пакетного зображення. Зразки RoI присвоюються класу, якщо їх перетин над об'єднанням із полем "Земна правда" перевищує 0,5. Інші RoI належать до фонового класу. Як і в класифікації, RoI з того ж зображення поділяють обчислення та використання пам'яті. Для збільшення даних вихідне зображення перевертається горизонтально з вірогідністю 0,5. Класифікатор softmax і регресори обмежувальної коробки налаштовуються разом за допомогою функції втрати з декількома завданнями, яка враховує як справжній клас відібраного RoI, так і зсув вибіркового обмежувального вікна від справжнього обмежувального поля.

### 1.3.2 Генерування та використання пропозицій регіону

Метою генерації пропозицій регіону при виявленні об'єктів є максимальне згадування, тобто генерування достатньої кількості регіонів для відновлення всіх справжніх об'єктів. Генератор менш стурбований точністю, оскільки завданням детектора об'єктів є ідентифікація правильних областей з виходу генератора пропозицій регіону. Однак кількість генерованих пропозицій впливає на ефективність роботи.

Існує два основні підходи до генерації регіону: генерація щільного набору та розріджена генерація.

Рішення щільних наборів намагаються генерувати грубою силою вичерпний набір обмежувальних коробок, що включає в себе кожне потенційне розташування об'єкта. Цього можна досягти, просунувши вікно виявлення по зображенню.

Однак пошук у кожному місці зображення обчислювально дорогий і вимагає швидкого детектора об'єктів. Додатково потрібно враховувати різні форми та розміри вікон. Таким чином, більшість методів розсувного вікна обмежують кількість об'єктів-кандидатів за допомогою грубого крокового розміру та обмеженої кількості фіксованих співвідношень сторін.

Більшість пропозицій регіону у щільному наборі не містять цікавих об'єктів. Ці пропозиції потрібно відкинути після фази виявлення об'єкта. Результати виявлення можуть бути відкинуті, якщо вони відстають від заздалегідь визначеного порогу довіри або якщо їх довірче значення нижче локального максимуму.

Замість того, щоб відкидати регіони після етапу виявлення об'єкта, сам генератор пропозицій областей може класифікувати регіони за класово-агностичним способом та відкидати низькопопулярні регіони. Це створює розріджений набір виявлення об'єктів. Подібно до методів щільного набору, після фази виявлення можуть бути реалізовані порогові показники та не локальний максимум для подальшого покращення якості виявлення. Різноманітні рішення можуть бути згруповані в методи без нагляду та нагляду.

Одним з найпопулярніших непідконтрольних методів є селективний пошук, який використовує ітераційне злиття суперпікселів. Існують також інші методи, що використовують той самий підхід [17]. Інший підхід – ранжувати об'єктивність розсувного вікна. Популярним прикладом цього є Edge Boxes [17], який обчислює оцінку об'єктивності, обчислюючи кількість ребер у обмежувальній коробці та віднімаючи кількість ребер, які перекривають межу поля. Існує також третя група методів, заснованих на сегментації насіння [17].

Контрольовані методи розглядають генерацію пропозицій у регіоні як проблему класифікації або регресії. Це означає використання алгоритму машинного навчання, такого як машина вектора підтримки [15]. Можна також використовувати згорткову мережу для створення регіонів, що цікавлять. Прикладом використання CNN для обчислення обмежувальних коробок є Multi-Box [16]. Деякі вдосконалені методи виявлення об'єктів, такі як Fast R-CNN [20], використовують частини тієї ж згорткової мережі як для створення пропозицій регіону, так і для виявлення. Ми називаємо такі методи інтегрованими методами.

Вибірковий пошук [11] використовує ієрархічний розподіл зображення для створення розрідженого набору об'єктів. Основна філософія дизайну полягає не у використанні єдиної стратегії, а в поєднанні найкращих особливостей сегментації знизу вгору та вичерпного пошуку. Автори мали три основні міркування: пошук повинен охоплювати всі масштаби, бути різноманітним, тобто не використовувати жодної єдиної стратегії групування регіонів та бути швидким для обчислення.

Алгоритм починається зі створення набору невеликих початкових областей за допомогою методу під назвою Сегментація зображень на основі графіків [18], розроблених Фельценцвальбом та Гуттенлохером. Метод створює набір областей, званих суперпікселями. Суперпікселі внутрішньо майже рівномірні. У поєднанні вони охоплюють усе зображення, але окремо вони не повинні охоплювати різні предмети. Потім селективний пошук продовжується ітераційним групуванням регіонів разом за допомогою жадного алгоритму, починаючи з двох найбільш схожих областей. Для обчислення подібності застосовується багато безкоштовних заходів. Ці заходи враховують кольорову схожість (за допомогою обчислення кольорової гістограми), схожість текстури (шляхом обчислення SIFT-подібної міри), розмір регіонів (малі регіони повинні бути об'єднані раніше) та наскільки добре регіони поєднуються (слід уникати прогалів). Фаза групування закінчується, коли кожен регіон був об'єднаний. Таким чином, створені гіпотетичні місця розташування об'єктів упорядковуються за ймовірністю



розташування, що містить об'єкт. На практиці місця впорядковуються виходячи з порядку, в якому вони були згруповані за різними заходами. Додано певний елемент випадковості, щоб запобігти занадто сильному вибору великих об'єктів. Дублікати нижчого рейтингу видаляються. Як метод генерування регіону, так і заходи подібності були обрані для швидкого обчислення, що робить метод в цілому швидким. Окрім використання різноманітних заходів подібності, пошук можна додатково урізноманітнити за допомогою додаткових кольорових просторів (для забезпечення інваріантності освітлення) та використання додаткових початкових областей.

Як впливає з назви, Edge Boxes [17] заснований на виявленні об'єктів з реберних карт. Основним внеском авторів методу є спостереження, що кількість реберних контурів, повністю закритих обмежувальним вікном, співвідноситься з ймовірністю того, що коробка містить предмет. По-перше, карта ребра обчислюється за допомогою методу тих же авторів, який називається Structured Edge Detector [15]. Потім товсті лінії ребер проріджують за допомогою локального максимуму. Замість того, щоб працювати на крайових пікселях безпосередньо, пікселі групуються за допомогою жадного алгоритму. Міра спорідненості розроблена для обчислення того, чи належать групи ребер до одного контуру. Пропозиції регіону знаходять шляхом сканування зображення за допомогою традиційного методу розсувного вікна та обчислення балів об'єктивності в кожній позиції, співвідношенні сторін та масштабі. Оцінка обчислюється шляхом підсумовування сили кромки реберних груп, які повністю лежать в коробці, і віднімання сили реберних груп, які є частиною контуру, що перетинає межу поля. Потім перспективні регіони ще більше удосконалюються.

### 1.3.3 Просунуте згорткове знаходження об'єктів

У цій дисертації ми зупинимося головним чином на Fast R-CNN. Однак існує кілька найсучасніших алгоритмів із покращеним часом чи точністю обчислень. Далі ми опишемо два з цих алгоритмів.

Faster R-CNN [10] є інтегрованим методом. Основна ідея полягає у використанні спільних згорткових шарів для створення пропозицій у регіоні та для виявлення. Автори виявили, що картки зображень, створені мережами виявлення об'єктів, також можуть використовуватися для створення пропозицій регіону. Повністю згорнута частина мережі швидшого R-CNN, яка генерує пропозиції щодо функцій, називається регіональною мережею пропозицій (RPN). Автори використовували архітектуру Fast R-CNN для мережі виявлення. Faster мережа R-CNN навчається шляхом чергування тренінгів для генерації та виявлення RoI. Спочатку навчаються дві окремі мережі. Потім ці мережі поєднуються та налагоджуються. Під час тонкої настройки певні шари зберігаються нерухомими, а певні шари навчаються по черзі.

Навчена мережа отримує єдине зображення як вхід. Спільно повністю згорнуті шари генерують карти зображень із зображення. Ці функціональні карти подаються в RPN. RPN виводить пропозиції регіонів, які вводяться разом із зазначеними картами функцій на кінцеві шари виявлення. Ці шари включають шар об'єднання RoI та виводять остаточні класифікації.

Використовуючи спільно розгорнуті шари, пропозиції регіонів обчислювально майже не вимагають витрат. Обчислення пропозицій регіону на CNN має додаткову користь від реалізації в GPU. Традиційні методи генерації RoI, наприклад, селективний пошук, реалізуються за допомогою процесора.

Для роботи з різними формами та розмірами вікна виявлення метод використовує спеціальні анкерні коробки замість піраміди зі зменшеними

зображеннями або піраміди різного розміру фільтрів. Якорні ящики функціонують як орієнтири для пропозицій різних регіонів, зосереджених на одному пікселі.

Детектор Multibox Single Shot [16] (SSD) ще більше забезпечує інтегральне виявлення. Метод взагалі не генерує пропозиції, а також не передбачає перекомпонування сегментів зображень. Він генерує виявлення об'єктів за допомогою одного проходу згорткової мережі.

Трохи нагадуючи метод розсувного вікна, алгоритм починається з набору за замовчуванням обмежувальних коробок. Сюди входять різні пропорції та масштаби. Прогнози об'єктів, обчислені для цих полів, включають параметри зміщення, які передбачають, наскільки правильне обмежувальне поле, що оточує об'єкт, відрізняється від поля за замовчуванням.

Алгоритм має справу з різними масштабами, використовуючи особливі карти з різних різних згорткових шарів (тобто великих і менших карт функцій) як вхід до класифікатора. Оскільки метод генерує щільний набір обмежувальних коробок, за класифікатором слідує стадія локального максимуму, що усуває більшість ящиків нижче певного порогового рівня довіри.

Вище ми описали, що Fast R-CNN швидший і точніший, ніж звичайний R-CNN. Але як працює Fast R-CNN порівняно з іншими передовими методами? Вчені порівнював показники швидких R-CNN, швидших R-CNN та SSD на тестовому наборі PASCAL VOC 2007. Під час використання мереж, підготовлених за навчальними даними PASCAL VOC 2007, Fast R-CNN досяг середньої середньої точності (mAP) 66,9. Fast R-CNN працювало трохи краще, з дозволом 69,9. SSD домігся дозволеного значення 68,0 з розміром входу 300 x 300 і 71,6 з розміром входу 512 x 512. Як стандартні реалізації швидкого R-CNN і швидшого R-CNN використовують 600 як довжину коротшого розміру вхідного зображення. SSD здається, ефективніші зображення із зображеннями однакового розміру. Однак для досягнення цього результату SSD вимагає широкого використання розширення даних [16].

Fast R-CNN та швидший RCNN використовують лише горизонтальне перевертання, і наразі невідомо, чи отримають вони користь від додаткового збільшення. Хоча вдосконалені методи більш точні, ніж Fast R-CNN, реальні поліпшення походять від швидкості. Коли більшість детектувань з низькою ймовірністю усуваються за допомогою порогового значення та локального максимуму, SSD512 може працювати на 19 FPS на GPU Titan X. Тим часом швидше R-CNN з архітектурою VGG-16 працює зі швидкістю 7 FPS [19].

Оригінальні автори Faster R-CNN [20] повідомляють про час роботи 5 FPS, тобто 0,2 с на зображення. Швидкий R-CNN має приблизно однакову швидкість оцінки, але вимагає додаткового часу для обчислення пропозицій регіону. Час генерації регіону залежить від методу, при цьому для селективного пошуку потрібні 2 секунди на зображення в процесорі та крайових полях, що вимагає 0,2 секунди на зображення [20].

#### 1.4 Постановка задачі

Метою роботи є надання можливості розпізнавати та класифікувати об'єкти за допомогою згорткових нейронних мереж. Для досягнення мети необхідно вирішити такі задачі:

1. аналіз можливостей згорткових нейронних мереж;
2. розробка алгоритму розпізнавання об'єктів на зображенні;
3. реалізація даного алгоритму та протоколу доступу.

Створений алгоритм повинен відповідати таким вимогам:

- якісно розпізнавати задані візуальні ключі;
- зрозумілий та чіткий дизайн API;
- простота в налаштуванні та використанні через API.

## Висновки по розділу

У першому розділі проведено аналіз існуючих рішень та предметної області, проаналізовані основні сучасні алгоритми та підходи, визначено переваги та недоліки цих рішень. В результаті проведеного аналізу сформульована постановка задачі, наведене призначення, цілі та задачі розробки.

## РОЗДІЛ 2. АНАЛІЗ ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

У цьому розділі ми розпочинаємо обговорення експериментальної частини дисертації. Спочатку, ми обговоримо критерії відбору методів та наборів даних. Тоді ми будемо описати вибрані методи, їх параметри та вибрані набори даних. Нарешті, ми обговоримо пост-обробку та оцінку. Впровадження з методів здебільшого обговорюється в наступній главі. Однак деякі деталі реалізації також обговорюються в цій главі, оскільки вони впливають на вибір методу.

### 2.1 Критерій вибору

Основними завданнями експериментальної частини були реалізація згорткового детектора об'єктів, перевірка наскільки добре детектор, навчений на загальних даних, виконує конкретну задачу та знайти потенційні джерела для вдосконалення. Ці завдання, а також практичне середовище реалізації визначали критерії відбору методів та наборів даних, що використовуються для експериментів. Основними проблемами, що обмежують використання методів глибокого навчання, є наявність обчислювальної потужності та даних про навчання. На цю тезу ми не мали доступу до ферми серверів або високоякісного GPU, використовуваного для наукових цілей. Швидше, нам потрібно було реалізувати методи на звичайному споживчому ноутбучі. Навчання мереж, що розвиваються, від початку до кінця на такому обладнанні, зайняло б багато часу. Таким чином, ми віддавали перевагу методам, які були створені достатньо, щоб мати доступні заздалегідь підготовлені мережі для використання в якості вихідної точки. Ми також віддавали перевагу методам, які мали доступ до реалізації MATLAB. Ми вирішили не проводити точну оцінку часу виконання та сконцентрувались на оцінці середньої точності.

Оцінка часу виконання вимагає більш стандартного середовища (а краще декількох різних комп'ютерів різного рівня продуктивності) для забезпечення

результатів з науковою цінністю. Однак ми повідомляємо, якщо знайшли деякі методи непрактично повільними. Окрім обчислювальної потужності, глибокі згорткові мережі вимагають великого обсягу навчальних даних. Оскільки збирання та анотування набору даних достатнього розміру є трудомістким, більшість досліджень проводиться на кількох доступних наборах даних. З тих же причин ми виключили збір даних для цієї тези. Натомість ми вирішили експериментувати з декількома різними доступними наборами даних.

Загальний детектор об'єктів повинен бути навчений на різних даних та підтримувати декілька категорій об'єктів. Стандартні набори даних орієнтирів (PASCAL VOC та ImageNet) забезпечили відповідну вихідну точку.

Загальний детектор об'єктів також повинен бути легкодоступним та легким у застосуванні. Існує багато доступних мереж, які перевірялися на стандартних наборах даних. Для оцінки працездатності загального детектора об'єктів у конкретному завданні ми зібрали дані тестів з різних джерел. Основним критерієм було те, що об'єктні анотації тестових даних повинні бути приблизно сумісні з об'єктними анотаціями стандартних орієнтирів. Це означає, що тестові набори даних повинні мати спільні класи зі стандартними орієнтирами і включати анотації об'єктів як обмежувальні поля.

Для пошуку потенційних вдосконалень згорткових систем ми реалізували геометричну модель оцінки сцени, опубліковану в статті "Об'єкти в перспективі" Хойем та ін.. Модель чітко налаштовує ймовірність виявлення на основі того, чи розташовані об'єкти в геометрично ймовірній частині зображення. Оскільки згорткові мережі в основному враховують місцеве сусідство об'єктів, розгляд всієї сцени забезпечує цікавий контраст і може потенційно покращити якість виявлення.

## 2.2 Знаходження об'єктів

Ми вибрали Fast R-CNN як основний метод для експериментів з виявлення об'єктів. Швидкий R-CNN вже налагоджений і має впровадження та попередньо перевірені мережі, доступні для декількох різних платформ. Незважаючи на те, що вдосконалені методи, такі як швидше R-CNN, забезпечують незначне підвищення точності, головний їх внесок – підвищення швидкості. Оскільки оцінка часу виконання була здебільшого поза межами даної тези, ці методи мали б мало додаткового значення для експериментів.

Через середовище впровадження (яке більш докладно описано в наступному розділі), ми вирішили пропустити початкову підготовку та використати перевірену мережу VGG-16. Наразі VGG-16 є однією зі стандартних.

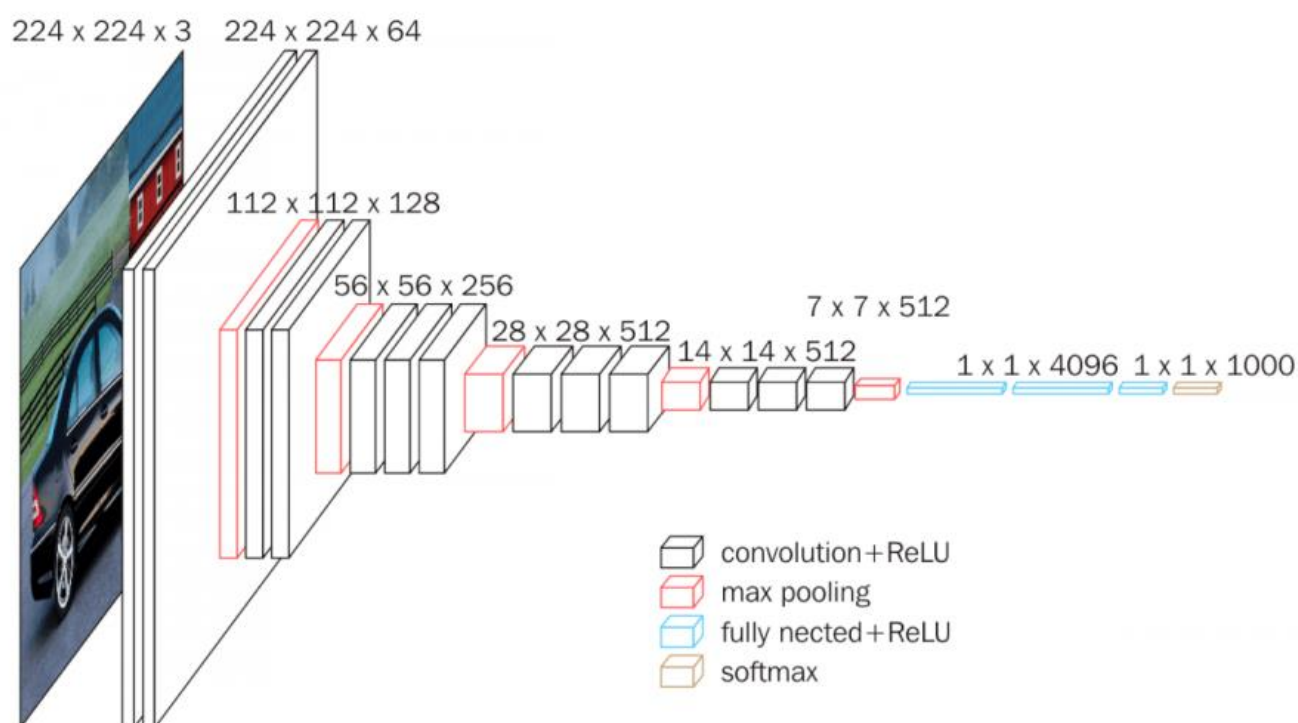


Рисунок 2.1. Структура мережі Fast R-CNN VGG-16

В оригінальній публікації Fast R-CNN [20] VGG-16 згадується як приклад "дуже глибокої мережі" і використовується як велика мережева модель (L) у всіх



експериментах. Ми використовували перевірену мережу з тією ж мережевою моделлю, структура якої показана на рис. 2.1. Більшість операцій мережі відображаються як окремі шари. Повністю згортова підмережа, показана починаючи вгорі зліва, включає чергуються згорткові шари та шари активації ReLu. Після кожні дві-три пари з них проводиться об'єднання. Розмір фільтра кожного згорткового шару становить  $3 \times 3$  (визначається як розмір ядра 3). Розмір ядра об'єднаних шарів становить 2. Скручені шари мають крок 1, а шари об'єднання мають крок 2.

Області, що цікавлять (RoI), вводяться на кінцевому шарі об'єднання повністю згорткової підмережі. Цей шар об'єднання RoI вибирає RoISpecific частину карти активації остаточного згорткового шару і виводить цю частину до повністю пов'язаної підмережі. Ця мережа включає два повністю пов'язані шари та два активаційні шари ReLu. Повністю пов'язані шари містять 4096 вихідних параметрів, що відображають розмір кінцевої карти активації. У шарах використовується регуляризація випадання з співвідношенням 0,5. Після другого шару активації обробка розгалужується для обчислення ймовірностей виводу та уточнення обмежувальної коробки.

#### 4.3 Генерування регіону

Оскільки для детектора об'єктів було обрано швидкий R-CNN, потрібні були також окремі методи генерації регіону. Оскільки це було досить просто здійснити, ми вирішили експериментувати як із селективним пошуком [11], так і з віконними полями [17]. Далі ми обговоримо, як були обрані параметри параметрів методів. Обидва способи мають параметри параметрів, які змінюють якість та кількість кандидатів у регіоні. Автори методів забезпечили ретельне вивчення впливу цих параметрів. Ми вибрали параметри на основі їх результатів, а не проводили власні експерименти з використанням набору перевірок.

## 2.3 Генерація регіонів

Оскільки Fast R-CNN був обраний в якості детектора об'єктів, також були потрібні окремі методи генерації регіону. Оскільки це було досить просто здійснити, ми вирішили експериментувати як із селективним пошуком [11], так і з віконними полями [17]. Далі ми обговоримо, як були обрані параметри параметрів методів. Обидва способи мають параметри параметрів, які змінюють якість та кількість кандидатів у регіоні. Автори методів забезпечили ретельне вивчення впливу цих параметрів. Ми вибирали параметри на основі їх результатів, а не проводили власні експерименти з використанням набору перевірок які змінюють якість та кількість кандидатів у регіоні. Автори методів забезпечили ретельне вивчення впливу цих параметрів. Ми вибирали параметри на основі їх результатів, а не проводили власні експерименти з використанням набору перевірок.

Селективний пошук може використовувати кілька стратегій диверсифікації. Згідно з оригінальним документом [11], додавання стратегій диверсифікації зазвичай дає кращі результати, але збільшує час обчислення. Оскільки в цій тезі час обчислення не оцінюється, ми вибрали перевагу якості. Ми вибрали режим налаштування, який автори називають «Якість вибіркового пошуку». Це включає використання всіх чотирьох стратегій диверсифікації та п'яти різних кольорових просторів: HSV, Lab, rgI (rg каналів нормованої RGB плюс інтенсивності), H (відтінок каналу від HSV) та I (тільки інтенсивність).

Параметри методу генерації суперпікселів [18], використовувані селективним пошуком, також впливають на вихід. Параметр  $k$  визначає кількість використаних початкових сегментів. Автори селективного пошуку встановили, що алгоритм був найбільш точним для виявлення об'єктів із значеннями  $k$  від 50 до 300. Ми вибрали середнє значення  $k = 200$ . Перед сегментацією суперпікселів зображення розгладжується за допомогою фільтра Гаусса, який контролюється через параметр  $\sigma$ . Оригінальні автори [18] завжди використовують  $\sigma = 0,8$ , що

вважає їх корисним для видалення артефактів оцифрування, не змінюючи зображення. Цей параметр також використовується тут.

Edge Boxes [17] знаходить початкових кандидатів у регіоні за допомогою пошуку розсувного вікна. Розмір кроків пошуку регулюється параметром  $\alpha$ , який визначає перетин над об'єднанням (IoU) сусідніх полів. Цей же параметр визначає розмір кроку для перекладу, масштаб та співвідношення сторін. Після того, як скриньки знайдені та уточнені, їх сортують та здійснюють локального максимуму. Під час NMS ящики викидаються, якщо їх IoU з вищою шкалою вище, ніж  $\beta$ . IoU та NMS пояснюються більш докладно у розділі 4.6.  $\alpha$  і  $\beta$  – найважливіші параметри Edge Boxes. Автори методу вивчали вплив цих параметрів з точки зору пошуку відповідної точності кандидата для об'єкта детектора. Навіть незважаючи на те, що значення IoU, що перевищує 0,5, зазвичай використовується як критерій оцінки продуктивності детектора об'єктів (як межа для визначення справжніх збігів), детектор об'єктів часто працює краще, якщо об'єкти-кандидати ближче відповідають справжнім об'єктам. Таким чином, цільовий IO повинен бути встановлений вище мінімально прийнятного IO. Автори визначають цільовий IO як  $\delta$  і проводили експерименти зі значеннями  $\delta$  0,5, 0,7 та 0,9.

Оскільки Fast R-CNN у будь-якому випадку обчислює уточнення обмежувальних коробок [20], ми вирішили, що підвищення  $\delta$  вище 0,9 для цих експериментів не потрібно. Внаслідок цього збільшення кількості кандидатів лише уповільнить алгоритм. В якості цільового IO ми вибрали більш практичне значення  $\delta = 0,7$ .  $\alpha$  і  $\beta$  встановлюються відповідно до  $\delta$ . Якщо потрібне більш високе значення  $\delta$ , вибираються  $\alpha$  і  $\beta$  для виведення щільніших наборів обмежувальних коробок навколо видимих об'єктів. Автори встановили, що значення  $\alpha = 0,65$  дало найкращі результати для  $\delta < 0,9$ . Оптимальне значення  $\beta$  визначалося як  $\delta + 0,05$ . Таким чином, ми вибрали  $\alpha = 0,65$  і  $\beta = 0,75$  як значення параметрів. Для мінімальної межі оцінки об'єктивності  $h$  в  $b$  (використовується для відкидання нецікавих вікон до етапу уточнення) ми використовували значення

за замовчуванням 0,01. Ми також використовували значення за замовчуванням 10000 як максимальну кількість об'єктних пропозицій.

## 2.4 Оцінка об'єктів в контексті зображення

Швидкий R-CNN здійснює класифікацію здебільшого на основі локального сусідства об'єкта. Частково це пояснюється методом використання пропозицій регіонів, а частково – властивою трансляцією згорткової мережі. Приймальні поля досягають свого максимального розміру в глибокому кінці мережі, де розмір карти активації зменшено за допомогою об'єднання та кроку. Після цього кінцеві повністю з'єднані шари дозволяють проводити відрахування протягом повних активацій. Однак повністю підключена підмережа, як правило, неглибока. У версії VGG-16 Fast R-CNN є лише два повністю пов'язані шари. За нашими міркуваннями, повністю пов'язані шари просто вчать поєднувати активації з різних частин підрозміру в об'єкти класи.

Згорткове регіональне виявлення об'єктів потенційно може бути вдосконалено іншими методами, які більш ретельно враховують всю сцену. Один із способів виявити сцену – це оцінити 3D-геометрію двовимірного зображення та використовувати цю модель для сегментації зображення на основні частини. У цьому розділі ми пояснимо, як ми поєднали виявлення сцени з згортковим виявленням об'єктів.

3D-моделі нагадують спливаючі книги дітей. Зображення поділяється на грубі сегменти (наземний, небосхильний і вертикальний), які потім вирізають і складають у текстуровані тривимірні площини. Ці площини можуть відображатися з трохи різних кутів, щоб змінити точку зору вихідного зображення. Для виявлення об'єктів нас більше цікавить сегментація зображення, а не показ сцени. Метод сегментації [20] заснований на машинному навчанні. Спочатку спочатку створюється набір суперпікселів за допомогою сегментації зображень на основі графіку [18] (яка також використовується селективним

пошуком). Суперпікселі об'єднуються в групи, які називаються сузір'ями. Дані тренінгу використовуються для оцінки того, які суперпікселі, ймовірно, належать до тієї ж міченої групи. Оскільки суперпікселі можна комбінувати кількома способами, утворюється кілька сузір'їв, що перекриваються. Мітка кожного суперпікселя (земля, вертикаль або небо) оцінюється шляхом обчислення ймовірності мітки кожного сузір'я, до якого належить суперпіксель, і шляхом обчислення, наскільки однорідними є ці сузір'я. Вертикальний клас додатково поділяється на лівосторонній, центральний, правосторонній, пористий і суцільний підкласи.



(А) Початкове зображення

(Б) сегментовані зображення

Рисунок 2.2. Приклад сегментації зображення за допомогою «геометричного контексту»

Публікація 2008 року "Поставлення об'єктів у перспективу" [16] (згадана нижче як OiP) Noiem et al. використовує згадану сегментацію геометрії як інструмент для виявлення об'єктів. Основна ідея методу – змінити ймовірність виявленого об'єкта, виходячи з того, наскільки геометрично вірогідним є об'єкт. Метод використовує навчальні дані для оцінки ймовірностей різних геометрій поверхні (складених з поверхні об'єкта та сусідніх поверхонь) за даними класів об'єктів. Метод також оцінює точку зору камери. Це означає оцінити висоту

камери у сцені та положення горизонту на рівнині зображення. Виходячи з них, можна обчислити висоту об'єктів на зображенні. Якщо відомі очікувані висоти класів об'єктів, ймовірності екземплярів об'єкта можна відрегулювати виходячи з того, чи потрапляють вони в очікуваний діапазон висоти.

Також можливо виявити, чи розташований об'єкт на первинній площині, і відповідно відрегулювати ймовірність (очікується, що пішоходи та автомобілі повинні торкатися землі, а не літати в небі). В сукупності геометрія поверхні та точка зору включають велику кількість інформації про ймовірні місця розташування об'єктів. В середину цих оцінювачів підключається детектор об'єктів. Взаємодія між точкою зору, виявленням об'єкта та геометрією поверхні враховується при обчисленні підсумкового умовиводу, який обчислюється за допомогою алгоритму поширення вірування.

Метод виявлення об'єктів, який використовується в роботі OiP, аналогічний методам, використовуваним у 2000-х роках, таких як [10], і заснований на особливостях, виготовлених вручну. Автори демонструють, що геометричні умовиводи значно підвищують точність цього методу. Замінивши метод виявлення об'єктів на швидкий R-CNN, ми можемо вивчити, як змінилося виявлення об'єктів за останні десять років і чи корисні геометричні умовиводи для поточного покоління детекторів об'єктів. Автори описують заміну детектора об'єкта іншим як "безболісне". В основному, виявлення потрібно перетворити у формат введення, використовуваний функцією геометричного виводу OiP (деталі цього перетворення обговорюються в наступному розділі). Замість того, щоб виконувати регулярний пошук локального, виявлення з високим перекриттям групуються перед геометричною фазою висновку. Регулярні NMS відкидають виявлення, що перекриваються, і зберігає лише виявлення з максимальною впевненістю. Зберігаючи всі виявлення, геометричний модуль виводу може

перемикати вибране обмежувальне поле на більш пізній момент. Ця операція групування вимагає виконання під час заміни детектора об'єктів.

## 2.5 Датасети

Далі ми обговоримо набори даних, що використовуються в експериментах. Спочатку ми обговоримо стандартні набори даних базових показників, які використовувались для підготовки попередньо перевіреного швидкого детектора об'єктів R-CNN. Потім ми обговоримо додаткові набори даних, які ми використовували для тестування детектора об'єктів на практиці.

Стандартні орієнтири, що використовуються для виявлення об'єктів, – це виклик даних PASCAL Visual Object Classes (VOC) [17] та набір даних великого масштабу візуального розпізнавання ImageNet [13].

Основним набором навчальних даних був набір даних PASCAL VOC. Виклик PASCAL VOC проводився щорічно між 2005 та 2012 роками. У 2007 році в набір даних було включено 9 963 анотованих зображення з 20 класами об'єктів (людина, 6 різних тварин, 6 різних предметів домашнього господарства та 7 різних транспортних засобів, включаючи автомобілі). Проблема 2007 року була останньою, коли оприлюднені ґрунтовні правдиві дані. До останнього року набір даних збільшився до 54900 зображень, приблизно половина з яких складала набори тестів та перевірки. Кількість та тип різних завдань також змінювалися щорічно, але, для цієї тези, нас цікавили лише дані виклику виявлення об'єктів.

Перевірена мережа PASCAL VOC, яку ми використовували, була ініціалізована в мережі, підготовленій на наборі даних ImageNet. Виклик ImageNet організовується щорічно з 2010 року. Виклик ґрунтується на підмножині бази даних ImageNet, яка наразі включає 14,197,122 зображення [1], з яких 1,034,908 містять анотації обмежувальної коробки. У 2012 році виклик було організовано одночасно з викликом PASCAL. З 2013 року виклик включає виклик виявлення об'єктів, який наразі [2] включає 200 повністю маркованих категорій (включаючи

ті ж категорії, що і набір PASCAL VOC). Набір для тренувань складається з 456 567 зображень, а валідаційний набір складається з 21 112 зображень.

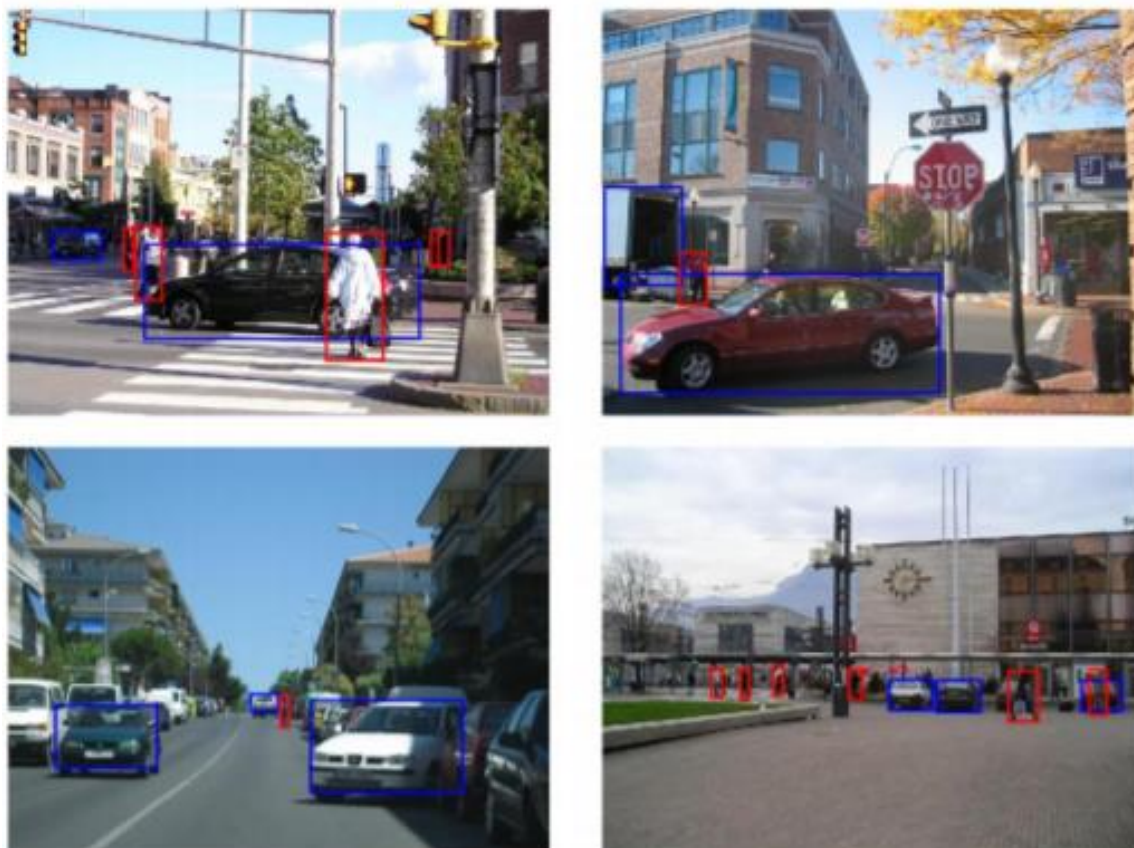


Рисунок 2.3. Приклад зображень з датасету з позначеними об'єктами

Виявлення об'єктів, пов'язаних з дорожнім рухом, таких як виявлення пішоходів та транспортних засобів – популярні дослідницькі теми в комп'ютерному зорі. Такі об'єкти помічені у багатьох загальнодоступних колекціях даних вулиць. Це дало чудове джерело даних для перевірки ввімкненого детектора об'єктів. Автомобілі та особи також розміщуються в наборах даних орієнтирів, що забезпечують взаємосумісність.

Перший тестовий набір даних, який ми використовували, – це той самий набір даних, який був використаний у публікації "Розміщення об'єктів у перспективі" (OIR) [16]. Набір включає 600 зображень тестового набору (422 з



яких включають дійсні об'єкти) та 60 зображень набору валідацій, вибраних із бази даних LabelMe. Об'єкти, що входять до зображень, позначені рамками з обмеженням. Ми використовували 60 зображень набору для перевірки як наш перший тестовий набір. Ця частина включає 126 автомобілів та 84 пішоходів.

Приклади даних OiP показані на рис. 2.3. Існує два об'єктні класи: автомобілі (включаючи звичайні легкові автомобілі, а також мікроавтобуси та вантажівки) та пішоходи (вертикальні особи). Набір даних PASCAL VOC включає приблизно відповідні категорії. Транспортні засоби набору даних PASCAL включають (пасажирські) автомобілі та автобуси як окремі класи, але ніяких інших чотириколісних транспортних засобів. Однак є обґрунтованою гіпотезою, що детектор, що навчається на цих двох класах, також навчиться розпізнавати більш загальний автомобільний клас набору даних OiP. Інший клас OiP, пішохід, суворо говорить про підклас категорії PASCAL. Можливі винятки включають велосипедистів та людей, що знаходяться всередині транспортних засобів чи будівель.



Рисунок 2.4. Приклади зображень із датасету "Сцени вулиць"

Набір даних Street Scenes [10], наданий Массачусетським технологічним інститутом, включає 3547 зображень, позначених для 9 різних об'єктів (автомобіль, пішохід, велосипед, будівля, дерево, дорога, небо, тротуар та магазин). Ми використовували 60 зображень, вибраних з цього набору, як наш другий тестовий набір. Ця частина набору включає 131 автомобіль та 51 пішохода.

Місцеположення об'єктів анотується за допомогою багатокутників, які щільно оточують об'єкт. Полігони можуть бути перетворені у форму обмежувального поля, знаходячи мінімальні та максимальні значення обох координат  $x$  та  $y$ . Автомобільні та пішохідні класи нагадують аналогічно названі класи OiP. До класу автомобілів належать усі моторизовані транспортні засоби (з більш ніж двома колесами) шириною більше 64 пікселів. До пішоходів належать особи, які ходять або стоять і мають висоту більше 32 пікселів. Обидва об'єкти маркуються лише у тому випадку, якщо вони видимі на 75%.

На рис. 2.4 наведено приклади даних вуличних сцен. Полігони перетворені на обмежувальні коробки. У верхньому лівому зображенні ми бачимо, що лише на більш великих автомобілях на передньому плані розміщено коментарі. На правому верхньому зображенні велосипедистка не позначається (оскільки вона не пішохід). У нижньому правому зображенні зазначаються лише вибрані особи із групи пішоходів.

## 2.6 Показники оцінки

Виявлення об'єктів оцінюється за допомогою стандартної метрики Перетину між Союзом (IoU) [17]. Обмежувальні коробки виявлення рідко відповідають ідеям пікселів і полям правдивості землі. На практиці нам цікаво знайти виявлення, які є досить близькими, щоб їх можна назвати справжніми позитивними збігами. IoU обчислюється діленням перетину (перекриття) вікна виявлення та поля земної правди на область їх об'єднання.

Взагалі оцінка IOU понад 0,5 зараховується як справжнє позитивне виявлення [17], і це визначення використовується і в цій тезі. У кожному полі основної істини може бути лише одна справжня позитивна відповідність. Якщо декілька виявлень відповідають одній основній істині, поле з найбільшою ймовірністю вибирається як справжнє позитивне відповідність, а інші виявлення позначаються як помилкові позитиви. Виявлення, що не відповідають рівню "земна правда", також позначені помилковими позитивами. Поле "Істина" без відповідних виявлень називається помилковим негативом.

Продуктивність алгоритму виявлення об'єктів оцінюється шляхом обчислення кривої точності виклику та інтерпольованої середньої точності виявлення для кожного класу, аналогічно виклику PASCAL VOC [17]. Ми обчислюємо криву безпосередньо для цілого набору даних (тобто над усіма об'єктами в тестових даних), а не обчислюємо середню криву, відібрану за певними значеннями виклику з кожного зображення. Ми створюємо комбінований перелік виявлень для кожного класу (з виявленнями, позначеними як істинні чи помилкові позитиви), відсортованих за оцінкою ймовірності класу. Точність та відкликання обчислюються сукупно у кожному місці виявлення у списку. Точність визначається як кількість отриманих справжніх позитивних виявлень, поділене на загальну кількість отриманих істинних та помилкових позитивних результатів. Нагадаємо, визначається як кількість отриманих справжніх позитивних виявлень, поділене на загальну кількість позитивних зразків, відомих з даних основної істини.

Інтерпольована (монотонно зменшується) крива обчислюється для цілей візуалізації та для обчислення інтерпольованої середньої точності. Інтерпольована крива створюється шляхом заміни фактичних значень точності на максимальні значення точності від решти частини кривої. Регулярно середня точність обчислюється обчисленням дискретного наближення площі під кривою точності відкликання (шляхом обчислення сукупної суми всіх точок, помножених на

дельта-значення відкликання). У виклику PASCAL VOC використовується замість середньої точності інтерполяція, яка зменшує вплив збурень, спричинених незначними різницями впорядкованості [17]. Ми використовуємо той самий метод, щоб отримати порівнянні результати. У наступних розділах термін середня точність слід розуміти як інтерпольовану середню точність.

Оцінки швидкого R-CNN не відкидаються на основі порогу ймовірності (як це було б у практичному застосуванні), оскільки нам цікаво дізнатися, чи досягне метод виявлення коли-небудь повного відкликання (як зменшується точність). Однак для швидкого виявлення R-CNN перед оцінкою проводять пошук локального максимуму (NMS). NMS виконується шляхом відкидання виявлень, які мають IoU більше заздалегідь заданого значення параметра з виявленням з більшою ймовірністю. Метою є видалення кількох виявлень одного і того ж об'єкта перед оцінкою. NMS виконується з використанням декількох значень параметрів для того, щоб знайти оптимальний поріг IoU.

Для геометричного умовиводу NMS не виконується як такий, як було пояснено раніше. Натомість виявлення з високим рівнем IoU групуються разом, і після виконання умовиводу вибирається найвище бальне виявлення групи. Ми також протестували різні значення цього параметра групування IoU.

### Висновки по розділу

В даному розділі розглянуто і проаналізовано способи та технології реалізації інтелектуального алгоритму, наведено опис і характеристики нейронних мереж, методів та датасетів, їх основні переваги та недоліки.

## РОЗДІЛ 3. ІМПЛЕМЕНТАЦІЯ АЛГОРИТМУ

У цьому розділі обговорюється практична реалізація Fast R-CNN та периферійних методів, а також проблеми, що виникають, та уроки отримані під час впровадження.

### 3.1 MatConvNet та Fast R-CNN

В даній дисертації розглянута реалізація через згорткову мережу за допомогою MatConvNet [3], що представляє собою інструментальний набір MATLAB, розроблений спеціально для цієї мети. Іншою альтернативою була б програма Caffe глибокого навчання [20], яка є більш популярною, універсальною та також має інтерфейс MATLAB. Однак MatConvNet забезпечив всю необхідну функціональність і простіший в установці.

MatConvNet – це сукупність функцій MATLAB, які реалізують різні будівельні блоки згорткової мережі. Обгортка CNN об'єднує ці блоки в повну мережу. Обгортка є або легкою обгорткою SimpleNN, або більш універсальною обгорткою DagNN.

Обгортка DagNN створює об'єкт MATLAB, який включає функції, що забезпечують доступ до навчання та оцінки мережі та параметрів, які зберігають мережеву структуру та ваги. Об'єкт можна зберігати на диску, полегшуючи обмін попередньо перевіреними мережами.

Веб-сайт MatConvNet пропонує перевірені мережі, які були перетворені у формат MatConvNet з оригінальних реалізацій і відповідають оригіналам до певної числової точності. Ми використовували реалізацію DagNN VGG-16 Fast R-CNN, яка була перевірена на наборі даних PASCAL VOC 2007 [17]. Мережа також використовує дані Imagenet Challenge 2012 [13], оскільки мережа Fast R-CNN була ініціалізована за допомогою мережі, підготовленої на даних ImageNet.

Реалізований сценарій MATLAB для виклику функцій CNN та для здійснення попередньої та післяобробки, що не передбачені MatConvNet. Сценарій починається з завантаження мережі, зображень та обмежувальних коробок з диска. Мережа переведена в тестовий режим, що відключає навчання. Потім обмежувальні поля перетворюються у формат координат, який використовується мережею. Далі зображення та поля масштабуються відповідно до вхідного розміру мережі (600x800 пікселів з 3 кольоровими каналами). Зображення також попередньо обробляються шляхом віднімання середнього значення кольору.

Після того, як дані будуть підготовлені, функція оцінювання мережі викликається, використовуючи зображення та поля в якості вхідних аргументів. Після оцінки ймовірності класу (softmax виходи) та значення дельти уточнення для кожного поля «видавлюються» з мережі. Далі уточнення обмежувальної рамки обчислюються комбінуванням вихідних значень і значень дельти. Удосконалені скриньки та ймовірності для цікавих класів (машини, автобуси та люди) зберігаються у структурі. Автомобільні та автобусні класи поєднуються в оцінці, щоб створити наближення до більш загального класу автомобілів, що включає більші автомобілі, а також легкові автомобілі. Нарешті, виявлення зберігаються на диску, а також проблеми, що виникають, та уроки, отримані під час впровадження.

### 3.2 Вибіркові поля пошуку та Edge Boxes

Автори оригінального документу публікують реалізацію MATLAB селективного пошуку [11]. Пакет включає в себе реалізацію C++ анізотропного фільтра гаусса та метод генерації суперпікселів [18]. Після їх компіляції додаткові налаштування не потрібні. Основна функція селективного пошуку вимагає, крім даних про зображення, певні початкові параметри в якості вхідних даних. Вони встановлюються, як пояснено в попередній главі. Після підрахунку початкових обмежувальних коробок дублікати видаляються за допомогою окремої функції.

Edge Boxes аналогічно доступний як код MATLAB методом [17] авторів. Додатково до основного коду потрібні панель інструментів для обробки зображень MATLAB та панель інструментів MATLAB для комп'ютерного зору Piotr [7]. Кілька функцій, які використовуються Edge Boxes, надаються лише у вигляді коду C ++, який компілюється під час налаштування. У пакет також входить реалізація детектору структурованого краю MATLAB (використовується для обчислення границь міцності та орієнтацій), з попередньо розробленою моделлю виявлення краю для RGB-зображень, яка підходить для наборів даних, використовуваних у цій дипломній роботі. Основна функція Edge Boxes приймає цю модель, параметри методу (пояснені в попередньому розділі) та зображення як вхідні дані.

### 3.3 Геометричні висновки

Реалізація методів MATLAB методів "Поставлення об'єктів у перспективу" (OiP) [5] та "Геометричний контекст" [11] доступна на сторінці [4]. Пакет Geometric Context включає реалізацію методу генерації суперпікселів на C ++ [18]. Реалізація OiP вимагає обох пакетів. Пакет OiP включає модифіковану версію автоматичного спливаючого коду для фотографій, який заміщений функцією обробки каталогів пакету Geometric Context. Ця функція вводить каталог файлів зображень, суперпікселі, обчислені для згаданих файлів, і перевірені класифікатори для сегментів зображення. Потім функція сегментує зображення в каталозі та зберігає їх у вихідному каталозі.

Фактичний геометричний висновок був реалізований шляхом зміни скрипту, включеного в пакет OiP. Сценарій спочатку завантажує сегменти зображення з диска і зберігає їх у структуру. Далі завантажуються ймовірності різних геометричних сегментів (заданого об'єкта). Вони передбачені в упаковці для автомобілів та пішоходів. Потім пріори точок зору ініціалізуються. Для цього ми використовували надані за замовчуванням розподіли (найімовірніше, висота

камери становить 1,67 м, а найімовірніше положення горизонту знаходиться в середині зображення).

Далі, Fast R-CNN висновки завантажуються з диска та перетворюються у формат «кандидати», використовуваний реалізацією OiP. Реалізація цього перетворення потрібна під час заміни детектора об'єктів, використовуваного оригінальним папером OiP. На практиці перетворення в основному включає копіювання виявлень у відповідний структурний формат. Як було пояснено в попередній главі, метод OiP передбачає, що не виявлено локального максимуму на виявленнях. Скоріше, як частина перетворення виявлення, розділи, що перекриваються 49, об'єднуються в об'єднане виявлення, яке спочатку має таку ж ймовірність, що і вікно, що обмежує найвищу довіру в групі. Вибір обмежувального поля може змінюватись як частина геометричного умовиводу.

Потім ми називаємо функцію умовиводу, надану в пакеті OiP. Ця функція приймає як введення виявлення у форматі кандидатів, сегментації зображень та пріорів точки зору. Функція виводить нових кандидатів та розподіл ймовірностей розташування горизонту та висот об'єкта. Ми перетворюємо нових кандидатів у формат виявлення, використаний раніше, і зберігаємо результати на диску, готові до оцінки.

### 3.5 Оцінка алгоритму

Сценарії для обчислення кривих точності відкликання та середньої точності здебільшого використовували наш власний код, за винятком NMS, який використовував реалізацію Томаша Малісевича, надану MatConvNet [3]. Подальші допоміжні функції також були написані для візуалізації зображення з його виявленнями та для друку помилкових позитивних та помилкових негативних виявлень. Деякі етапи обробки залежать від набору даних. Наприклад, за даними OiP, машини мають мінімальну висоту 14 пікселів, а пішоходи мають мінімальну висоту 36 пікселів. У наборі даних "Сцени вулиць" автомобілі мають мінімальну



ширину близько 64 пікселів, а людина має мінімальну висоту 32 пікселів. Виявлення, менші за них, були відкинуті.

Оскільки глибоке навчання – це сфера, що виникає, не існує простих «нестандартних» рішень для впровадження. Більшість корисних інструментів призначені для фахівців та дослідників. Установка та тестування цих інструментів перед реальною реалізацією поставила додаткову проблему.

Реалізація прискорення графічного процесора за допомогою MatConvNet вимагає, щоб панель інструментів була встановлена разом із CUDA, яка є платформою паралельних обчислень, розробленою NVIDIA. CUDA може бути додатково доповнена cuDNN, що є бібліотекою глибокої нейронної мережі CUDA. MatConvNet потребує графічного процесора з обчислювальною здатністю (вимірювання, що використовується NVIDIA для ранжирування їх графічного обладнання для використання CUDA) більше 2,0 [3]. GeForce 840M, що використовується для реалізації, має обчислювальну здатність 5,0 [6].

Таким чином, можна було б обчислити (принаймні згорнуту частину) трубопроводу для виявлення об'єктів на GPU. Однак складання для графічних процесорів та налаштування CUDA виявилось багато часу. Однією з особливих проблем були взаємні вимоги до версій, які мали різні інструменти (включаючи MATLAB). З іншого боку, подібні вимоги до версії завадили подальшим експериментам з Caffe.

Оскільки важко було передбачити, скільки потрібно додаткової роботи по налаштуванню прискорення GPU, і оскільки оцінка виконання в будь-якому випадку не виходить за межі цієї дипломної роботи, ми вирішили виконати всі обчислення на процесорі, як тільки ми створили робочу систему. Зрештою, CNN не був найбільш обчислювальною частиною експериментів.

Найбільш трудомісткою частиною було обчислення сегментів геометрії, що вимагало приблизно 10-20 хвилин для кожного зображення. Для даних вуличних сцен необхідно було зменшити масштаб оригінальних зображень 1280x960 до

640x480, оскільки, використовуючи вихідний розмір зображення, сегментація, як правило, ніколи не припиняється і не займає декілька годин. Масштабування масштабів, імовірно, не впливало на продуктивність виявлення, оскільки зменшені масштаби зображення були близькими до вхідного розміру CNN.

Незначною проблемою було перетворення зображень та обмежувальних коробок між різними типами даних та форматами. Наприклад, потрібно було масштабувати зображення до розміру входу мережі VGG-16. Оскільки обмежувальні поля були створені для оригінального розміру зображення, їх потрібно також масштабувати. Поле, сформоване селективними полями пошуку та окантовки, було несумісне між собою та не сумісне з основними полями правди та вхідними типами функцій, що використовуються для обчислення IoU та NMS. Індивідуально такі перетворення були тривіальними, але разом вони займали напрочуд велику кількість часу.

### 3.6 API дизайн та взаємодія з мобільним додатком

Важливою деталлю даної системи є можливість її взаємодії з мобільним додатком. Дана можливість може бути реалізована за допомогою розгортання API на базі будь-якої сучасної мови програмування, наприклад C# або Java. Дана технологія вибрана в зв'язку з тим, що це є типове рішення для побудови протоколу доступу в різних варіаціях. Система реалізовує протокол обміну даними REST.

REST це підхід до архітектури мережевих протоколів, які забезпечують доступ до інформаційних ресурсів. Був описаний і популяризований 2000 року Роєм Філдінгом, одним із творців протоколу HTTP. В основі REST закладено принципи функціонування Всесвітньої павутини і, зокрема, можливості HTTP. Філдінг розробив REST паралельно з HTTP 1.1 базуючись на попередньому протоколі HTTP 1.0.

Дані повинні передаватися у вигляді невеликої кількості стандартних форматів (наприклад, HTML, XML, JSON). Будь-який REST протокол (HTTP в тому числі) повинен підтримувати кешування, не повинен залежати від мережевого прошарку, не повинен зберігати інформації про стан між парами «запит-відповідь». Стверджується, що такий підхід забезпечує масштабовність системи і дозволяє їй еволюціонувати з новими вимогами.

Антиподом REST є підхід, заснований на виклику віддалених процедур (Remote Procedure Call, RPC). Підхід RPC дозволяє використовувати невелику кількість мережевих ресурсів з великою кількістю методів і складним протоколом. При підході REST кількість методів і складність протоколу суворо обмежені, що призводить до того, що кількість окремих ресурсів має бути великою.

REST, як і кожен архітектурний стиль відповідає ряду архітектурних обмежень (англ. architectural constraints). Це гібридний стиль який успадковує обмеження з інших архітектурних стилів.

Перша архітектура від якої він успадковує обмеження це клієнт-серверна архітектура. Її обмеження вимагає розділення відповідальності[en] між компонентами, які займаються зберіганням та оновленням даних (сервером), і тими компонентами, які займаються відображенням даних на інтерфейсі користувача та реагування на дії з цим інтерфейсом (клієнтом). Таке розділення дозволяє компонентам еволюціонувати незалежно.

Наступним обмеженням є те, що взаємодії між сервером та клієнтом не мають стану, тобто кожен запит містить всю необхідну інформацію для його обробки, і не покладається на те, що сервер знає щось з попереднього запиту.

Відсутність стану не означає що стану немає. Відсутність стану означає, що сервер не знає про стан клієнта. Коли клієнт, наприклад, запитує головну сторінку сайту, сервер відповідає на запитання і забуває про клієнта. Клієнт може залишити сторінку відкритою протягом кількох років, перш ніж натиснути посилання, і тоді

сервер відповість на інший запит. Тим часом сервер може відповідати на запити інших клієнтів, або нічого не робити. Для клієнта це не має значення.

Таким чином, наприклад дані про стан сесії (користувача, який автентифікувався) зберігаються на клієнті, і передаються з кожним запитом. Це покращує масштабовність, бо сервер після закінчення обробки запиту може звільнити всі ресурси, задіяні для цієї операції, без жодного ризику втратити цінну інформацію. Також спрощується моніторинг і зневадження, бо для того аби розібратись, що відбувається в певному запиті, досить подивитись лише на той один запит. Збільшується надійність, бо помилка в одному запиті не зачіпає інші.

Мінусом цього обмеження є те, що знижується продуктивність через те, що в кожен запит тепер доводиться додавати дані сесії з клієнта. Також збереження стану на різних клієнтах важче підтримувати, бо реалізації клієнтів можуть відрізнятись, тоді як середовище сервера повністю під контролем розробника.

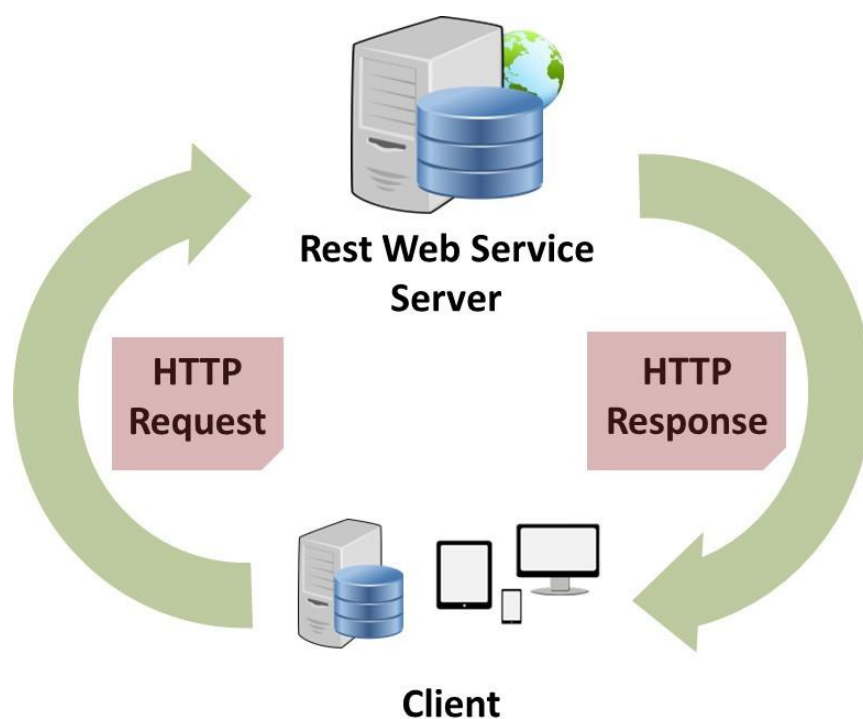


Рисунок 3.1. Типова REST Архітектура

Додатковим обмеженням стилю REST є те, що системи, написані в цьому стилі, повинні підтримувати кешування, тобто дані, які передаються сервером, повинні містити інформацію про те, чи можна їх кешувати, і якщо можна, то як довго. Це дозволяє збільшувати продуктивність, уникаючи зайвих запитів, але також зменшує надійність системи, через те, що дані в кеші можуть бути застарілими.

В системі є два основних ендпоінта: Post метод Configure та Get метод GetScore. Метод Configure використовується для налаштування конкретного візуального ключа, на який зараз орієнтується система та отримує числове значення даного ключа. Метод GetScore використовується для оцінки присутності конкретного візуального ключа на зображенні. Отримує як параметр файл зображення та віддає відповідь в форматі True/False відповідно до того, чи є даний візуальний ключ на зображенні чи ні.

Даний дизайн API цілком достатній для зручної роботи розробників мобільних додатків. Дане рішення дає можливість не обмежувати варіанти реалізацій мобільних додатків, тому що спілкування через REST є звичайною справою для архітектури мобільних застосунків.

Для тестування системи був розроблений невеликий мобільний застосунок для операційної системи Android. Тестування довело, що даний дизайн API цілком покриває потреби.

### 3.7 Вимоги до платформи

Зазначений алгоритм розроблявся на наступній платформі:

- процесор – n Intel Core i5-4210U 1.70 GHz CPU;
- оперативна пам'ять 8024 Мб;
- відеокарта NVIDIA GeForce 840M GPU;
- операційна система: Windows або Linux;

- доступ до мережі Інтернет.

### 3.8 Вимоги до платформи розміщення API

Надано вимоги для роботи з light-weight API

- БД Postgres 11 і вище;
- .NET середовище;
- Підтримка .NET Core;
- операційна система: Windows;
- процесор – n Intel Core i5-4210U 1.70 GHz CPU;
- оперативна пам'ять 8024 Мб;
- відеокарта NVIDIA GeForce 840M GPU;
- доступ до мережі Інтернет.

Висновки по розділу

В третьому розділі описано етап розробки алгоритмічного та програмного забезпечення системи. Проведений аналіз потрібних вимог системи та складені рекомендовані характеристики системи як для нейронної мережі, так і для API. Наведено архітектуру алгоритму та технічні вимоги, описано дизайн API та проведене тестування.

## РОЗДІЛ 4. ОЦІНКА АЛГОРИТМУ

У цьому розділі ми представимо результати експериментів. По-перше, ми надамо загальні результати із Fast R-CNN та порівняємо два методи генерації регіону. Потім ми обговоримо ефекти встановлення параметрів пошуку локальних максимумів. Також оцінимо та обговоримо метод геометричних висновків.

### 4.1 Fast R-CNN з різними регіонами

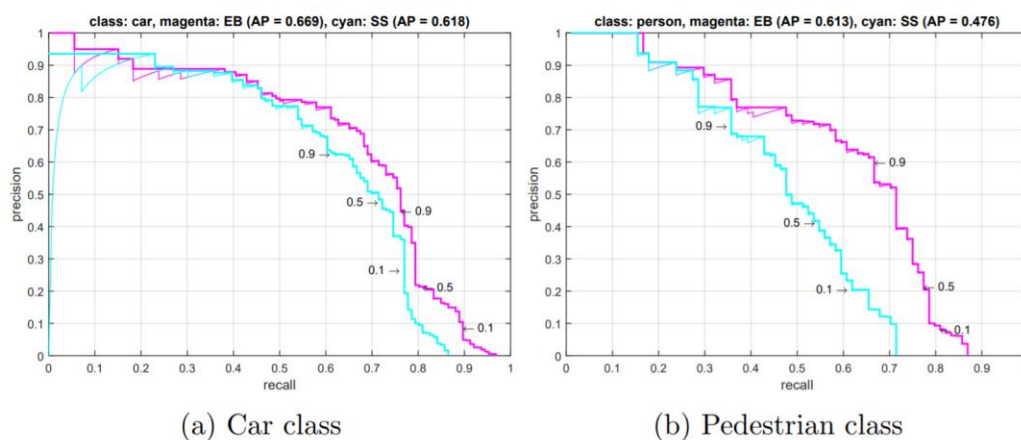


Рисунок 4.1. Криві точності

На рис. 4.1 показані криві точності виклику та середні точності швидких виявлень R-CNN набору даних OiP. Перед оцінкою було проведено пошук локального максимуму з  $\text{IoU} = 0,45$ . Виявлення обчислювались за допомогою двох різних методів генерації регіону: селективного пошуку (блакитного кольору) та крайових коробок (пурпурного). Як ми бачимо, Edge Boxes працює краще для обох класів (підвищення середньої точності на 0,0510 для автомобілів та 0,1370 для пішоходів). Однак зауважте, що обидва способи використовують попередньо обрані параметри. Цілком можливо, що точна настройка параметрів дала б кращі результати для одного або обох методів.

Загалом, результати виявлення Fast R-CNN є досить хорошими, маючи на увазі, що згорткова мережа була підготовлена на зовсім іншому наборі даних, який не включає багато навчальних зображень, пов'язаних із трафіком. Для автомобілів до 40% правильних екземплярів відкликаються з 90% точністю. Якби метод був використаний у практичному застосуванні, нам потрібно було б вибрати деякий поріг довіри, над яким ми розглядаємо виявлення як надійні. Три приклади порогових розташувань (0,1, 0,5, 0,9) показані поруч із кривими. Як ми бачимо, Edge Boxes забезпечує більше результатів у кожному конкретному випадку, з кращим відкликанням, але меншою точністю. Ймовірно, це пов'язано з тим, що Edge Boxes повертає ще багато кандидатів (у середньому 8000 коробок для зображень OiP проти 1000 коробок, повернутих селективним пошуком).

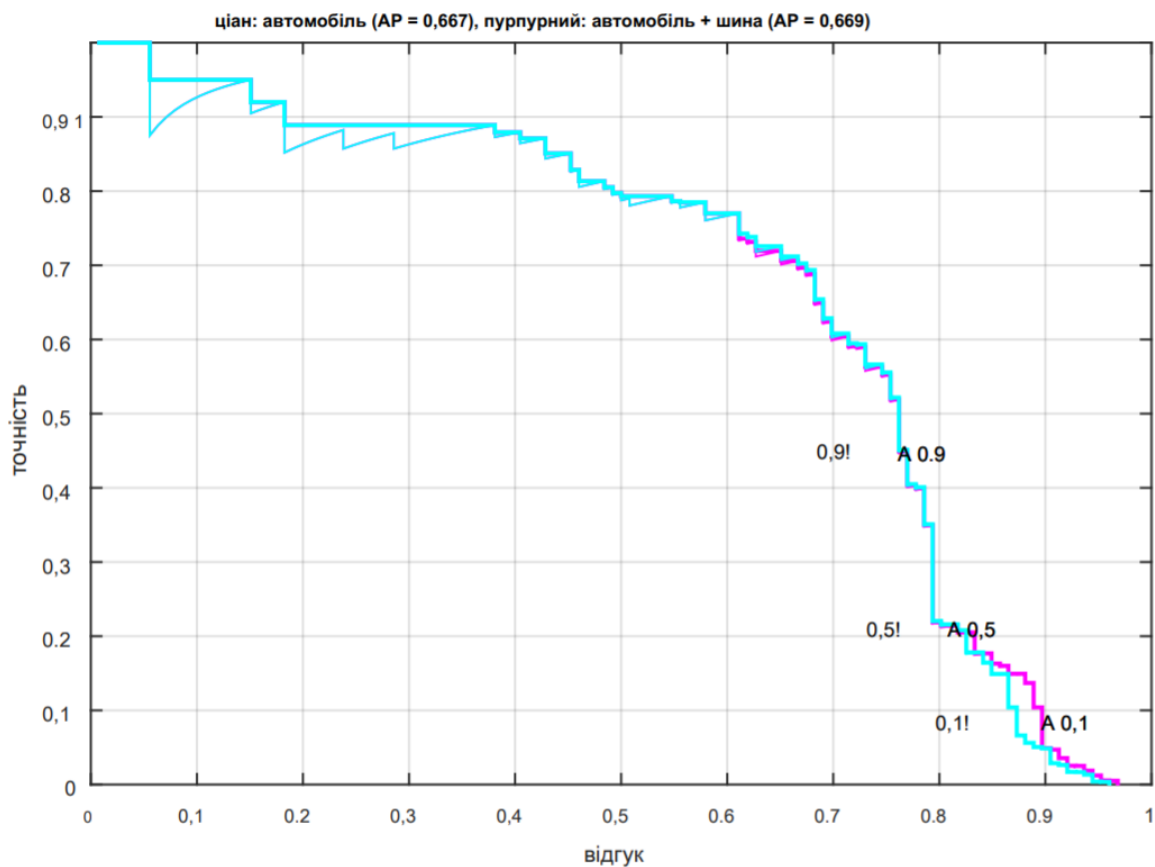


Рисунок 4.2. Додавання класу автобусів до виявлення автомобілів трохи покращує точність



Як було пояснено вище, виявлення для автомобілів – це комбінація автомобільних та автобусних класів PASCAL VOC, оскільки автомобілі визначені ширше у тестових наборах даних. Відповідні криві точності відкликання відображені на рис. 4.2. Ми можемо побачити, що детектор шини має ефект лише у важкому кінці кривої. Тим не менш, відмінності показують, що різниці між пасажирськими вагонами та автобусами розуміються детектором об'єктів і достатньо істотні, щоб змінити значення, навіть якщо воно незначне.

#### 4.2 Оцінка коректності роботи

На рис. 4.3 показано перші 50 виявлених помилково позитивних осіб із набору даних OiP (за допомогою Edge Boxes та Fast R-CNN)

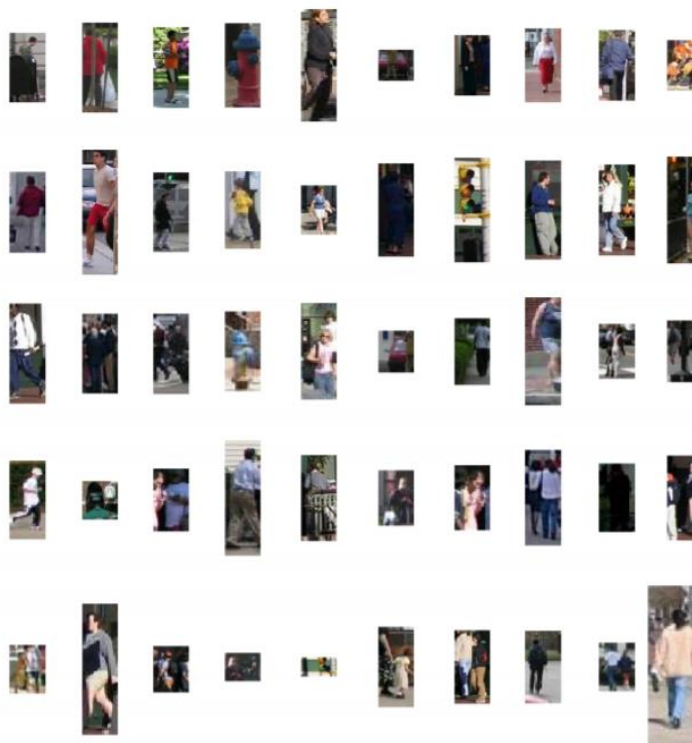


Рисунок 4.3. Топ-50 виявів помилкових позитивних даних із набору даних OiP

За допомогою візуального огляду близько 40 коробок містять людей. Решта це водяні пости, дорожні знаки, шишки та невідомі предмети. Мабуть, швидкий R-CNN працює значно краще, ніж вказує крива точності відкликання. Об'єктний детектор виявляє людей, що не були помічені. Більшість із них це пішоходи, які частково оклюзоровані або стоять занадто близько до інших людей. Наявність нелюдських об'єктів, ймовірно, пояснюється відсутністю важких негативних прикладів.



Рисунок 4.4. False негативні випадки з набору даних OiP

Відповідні помилкові негативні виявлення відображені на рис. 4.4. Це помічені пішоходи, яких не знайшов Fast R-CNN. Деякі випадки це люди, що стоять у тіні. Ці зразки важко виявити з повного зображення навіть людиною. Кілька інших це висококонтрастні предмети, що стоять перед аналогічним кольором фону.

Для останніх двох немає чіткого негайного пояснення того, чому алгоритм їх пропустив. Результати для автомобільного класу аналогічні. Для стислості вони тут не представлені.

### 4.3 Пошук локальних максимумів

На рис. 4.5 показаний ефект від налаштування строгості пошуку локального максимуму. Криві відображають середню точність виявлення автомобіля з даних ОіР. На правому кінці кривої, де  $\text{IoU} = 1$ , ми взагалі не виконуємо NMS (за винятком видалення точних дублікатів, що повинно було бути вже). Коли ми рухаємося на кривій ліворуч, параметр  $\text{IoU}$  зменшується, а це означає, що ми починаємо видаляти виявлення зі зменшенням перекриття з виявленням високої надійності.

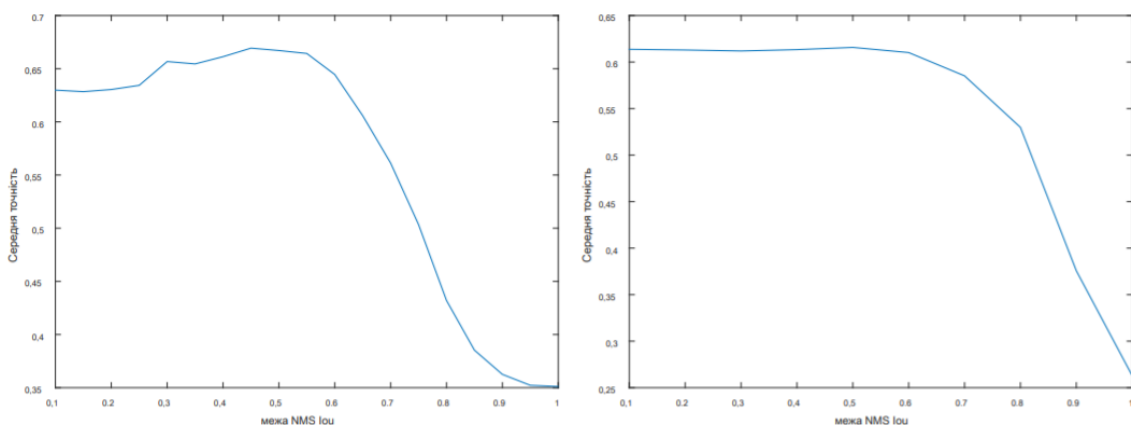


Рисунок 4.5. Ефект зміни параметра NMS IoU

Ми можемо бачити, що Edge Boxes отримує вигоду від NMS з "обох сторін", оскільки пошук локального максимуму та мінімуму погіршує результати. З іншого боку, селективний пошук показує поліпшення, починаючи з приблизно  $\text{IoU} = 0,6$ , але після цього зменшення значення параметра не забезпечує значних змін точності. Можливо, це пов'язано з тим, що селективний пошук дає менше результатів і менш успішний. При  $\text{IO} = 0,6$  більшість відволікаючих вторинних виявлень було видалено. Знижуючи межу далі, НМС теоретично повинен у певний момент почати видаляти перекриття справжніх позитивних виявлень та знижувати точність. Однак у результатах селективного пошуку ці поля чомусь відсутні.

Результати виявлення людини були аналогічні. На основі цих результатів для оптимальних значень NMS для експериментів було обрано  $\text{IoU} = 0,45$ .

#### 4.4 Геометричні висновки

На рис. 4.6 та рис. 4.7 показано вплив описаного вище методу геометричних висновків на виявлення об'єкта. На рис. 4.6 показана крива точності виклику швидких детектувань R-CNN (червона) даних OiP порівняно з кривою тих самих результатів після геометричного висновку (синій). На рис. 4.7 показано подібне порівняння для даних вуличних сцен.

Як зазначено вище, виявлення швидких R-CNN було оброблено за допомогою NMS з порогом  $\text{IoU} = 0,45$ . Оскільки метод геометричного висновку не використовує звичайні NMS, порівняно з використанням однакового порогового значення для обох методів не має сенсу проводити порівняння. Натомість ми обирали оптимальні порогові значення як для НМС, так і для методу групування індивідуально та порівняли результати.

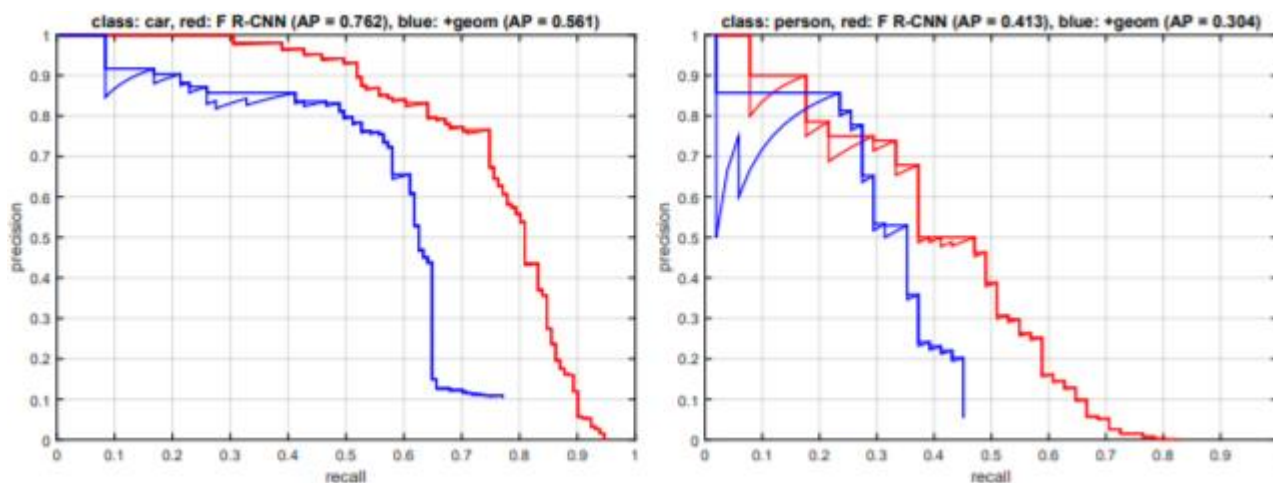


Рисунок 4.6. Порівняння простого Fast R-CNN (червоного) та OiP + Fast R-CNN (синього) за допомогою даних набору OiP

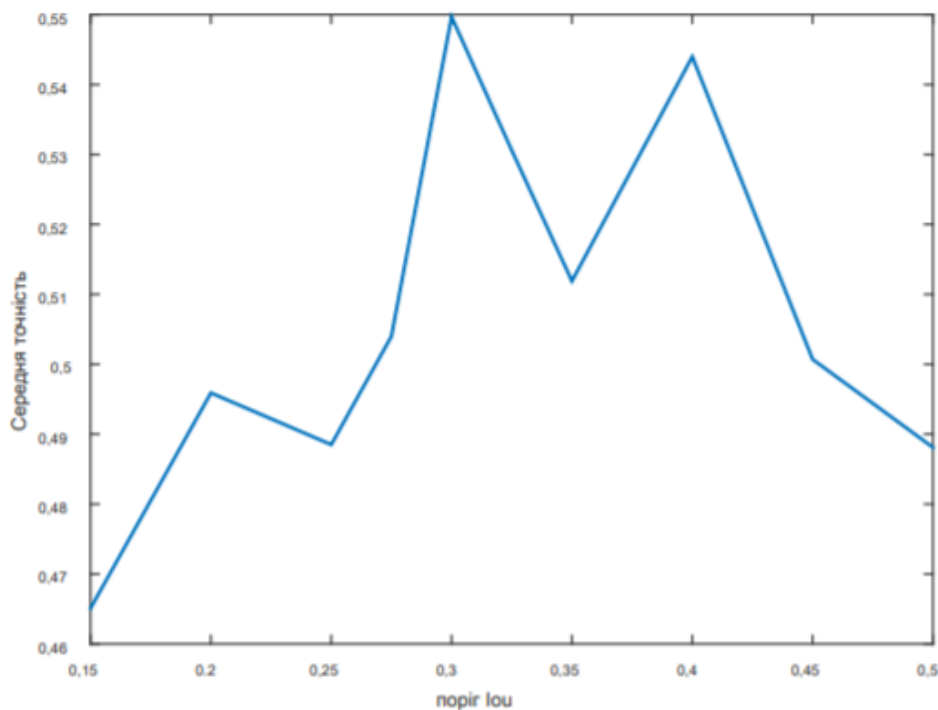


Рисунок 4.7. Порог IoU

На рис. 4.7 показано, як змінюється точність виявлення людини геометричним методом, коли ми змінюємо поріг IoU групування. Виходячи з цих результатів, ми вибрали 0,3 як оптимальний поріг групування, і криві точності відкликання були намальовані за допомогою цього значення. З міркувань продуктивності швидкі виявлення R-CNN з вірогідністю менше 0,01 відфільтровували перед геометричними висновками. Це пояснює, чому крива трохи зупиняється перед швидкою кривою R-CNN.

Криві точності виклику чітко показують, що принаймні на цих наборах даних та анотацій геометричні умовиводи здебільшого погіршують продуктивність Fast R-CNN. Виявлення автомобіля набору даних OiP це єдиний випадок, коли синя крива спочатку вище червоної кривої, це означає, що найбільш надійні результати краще після геометричного висновку. Зауважте тут, що оскільки метод групування, схожий на NMS, може змінити вибір обмежувальної коробки, найкращі результати не обов'язково полягають у тих самих полях.

Однак за допомогою візуального огляду окремих результатів ми можемо чітко знайти приклади роботи геометричних висновків за призначенням. Тут виявлення виявляються лише у тому випадку, якщо їх вірогідність перевищує 0,5. Поле правди наземне відображається темнішим кольором. На верхньому зображенні (взятому з набору даних OiP) ми бачимо, як геометричний висновок правильно знижує показник брезенту (висить збоку будівлі) нижче порогу. Деякі машини та самотній пішохід виявляються правильно обома методами. Жовтий автомобіль не виявляється жодним методом, ймовірно, тому що, будучи окульованим іншими об'єктами, він не виявляється в першу чергу як кандидат-об'єкт.

Однак геометрична модель також створює неповні додаткові виявлення автомобілів, які не зазначаються в оригінальних даних (насправді на зображенні не так багато машин). Це ілюструє типову поведінку геометричного методу. Fast R-CNN вже досить добре працює у визначенні класів об'єктів, але через оклюзії та неповних анотацій існує багато помірно ймовірних часткових виявлень, які не вважаються справжніми позитивними оцінками. Якщо ці виявлення також можуть бути геометрично правдоподібними, метод геометричних висновків ще більше збільшує їхню ймовірність. Отже, хоча метод правильно демонструє деякі помилкові позитивні виявлення, в той же час він сприяє багатьом іншим, які геометрично знаходяться в правильному місці, але насправді є некоректними (наприклад, дорожні знаки та водні пости, помилково сприйняті для людей), частина об'єктів, який уже виявлений або взагалі не позначений. Наявність великих груп чи ліній людей та автомобілів часто призводить до виявлення, які є частковими (оскільки вони заткнуті) та не зазначаються в початкових даних, і це сприяє геометричній моделі.

З кращими примітками ми могли б виконати більш змістовне порівняння. На нижньому зображенні на рис. 4.8 показано кілька цікавих випадків проблем, викликаних невідповідностями у визначеннях класу. Перший приклад –

велосипедист, якого не зараховують до пішохода (як уже згадувалося). На зображенні також є ще одна непомічена людина, яка стоїть за відкритими дверима вантажного автомобіля. Швидкий R-CNN (досить примітно) дійсно виявляє його як особистість, але геометричний модуль виведення демонструє виявлення, імовірно, тому, що людина не стоїть на землі (працює за призначенням і, на цей раз, відповідає анотації).

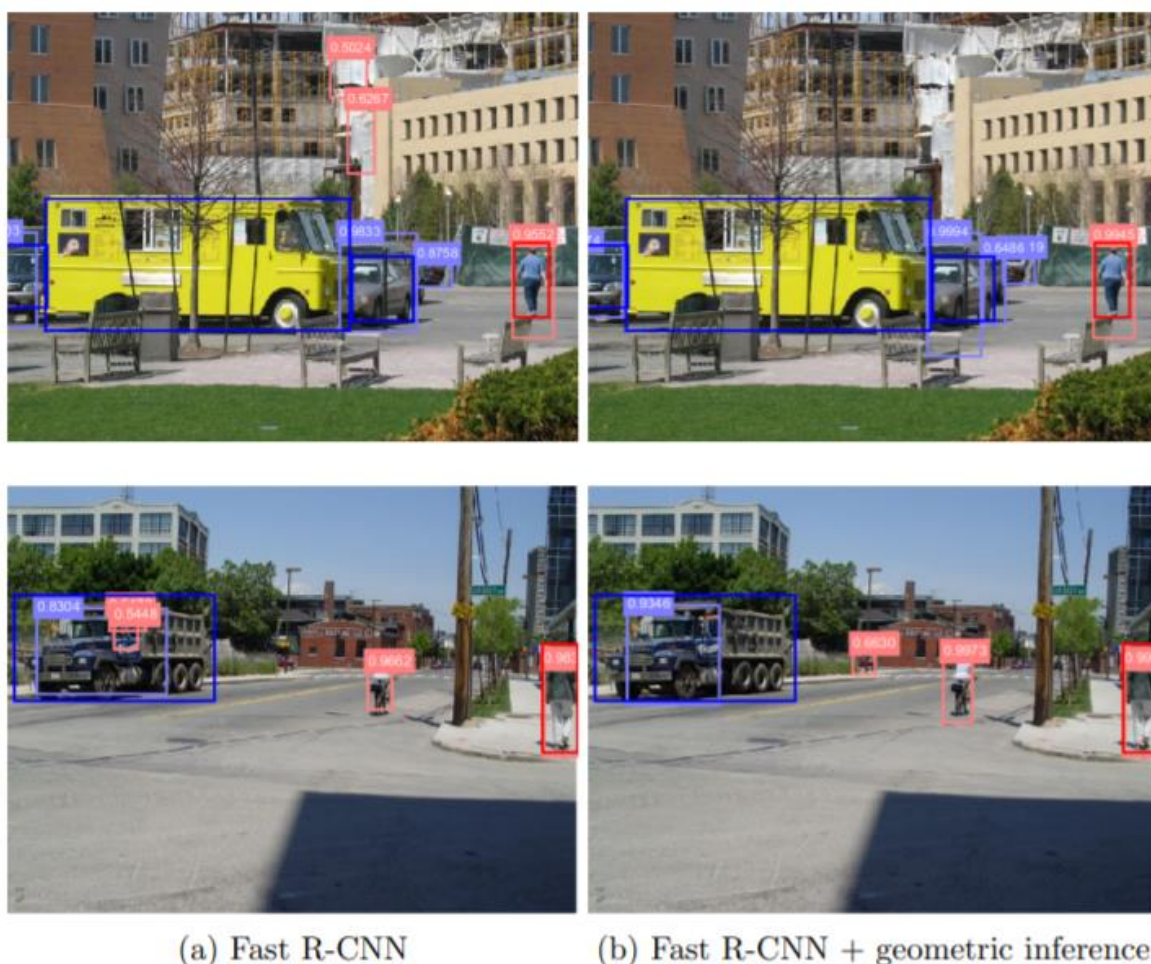


Рисунок 4.8. Приклади виявлення

Також обидва способи виявляють лише передній кінець вантажного автомобіля та не беруть до уваги на менший автомобіль. Це пояснюється тим, що виявлення автомобілів – це комбінація виявлення пасажирських автомобілів та



автобусів. Далі на задньому плані геометричний умовивід неправильно позначає оклюзований автомобіль як людину. Висота предмета, мабуть, підходить людині.

Загалом, це тип зображення, який дуже важко класифікувати однозначно, враховуючи незначні відмінності у визначенні класів. На щастя, більшість зображень у наборах даних містять чіткіші приклади різних класів, що робить можливим змістовне оцінювання результатів. Коротше кажучи, метод геометричного висновку не покращує середні результати швидкого R-CNN "поза коробкою", але певні випадки, коли метод працює, вказують на вдосконалення, які могли бути впроваджені або в згортковому виявленні об'єкта, або в геометричному умовиводі. У наступній главі ми обговоримо різні вдосконалення виявлення згорткових об'єктів.

### Висновки по розділу

В четвертому розділі описано етап оцінки алгоритмічного забезпечення, проведений аналіз результатів роботи алгоритму. Наведені приклади роботи алгоритму. Надані загальні результати із Fast R-CNN та порівняно два методи генерації регіону.



## РОЗДІЛ 5. МАРКЕТИНГОВИЙ АНАЛІЗ СТАРТАП-ПРОЕКТУ

### 5.1 Опис ідеї проекту

Таблиця 5.1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
Розробити систему для розпізнавання візуальних ключів	Система розпізнавання візуальних ключів	Зручний доступ до інформації, можливість використання в AR застосунках
	Організація великих подій, інтерактивні дороговкази	

Таблиця 5.2. Опис ідеї стартап-проекту

№	Техніко-економічні характеристики ідеї	Продукція конкурентів				W (слабка сторона)	N (нейтральна сторона)	S (сильна сторона)
		Мій проект	AlwaysAI	AWSImage	Cogniak			
1	Ліцензія	+	+	+	+	Потребує інвестицій на старті	Стабільність виплат	Можна отримувати дохід з підписок
2	Легкість в інтеграції	+	+	-	-	Обмеженість функціоналу	Підтримка системи	Привабливість для невеликих команд та компаній з обмеженими задачами

## 5.2 Технологічний аудит ідеї проекту

Таблиця 5.3. Технологічна здійсненність ідеї проекту

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1	Модуль нейронної мережі	Згортова нейронна мережа	Необхідно розробити	Доступна
2	API	Бекенд програма створена за допомогою .net Core web api, REST	Необхідно розробити	Доступна
3	Середовище розгортки	Docker	Наявна	Доступна
4	Рівень даних	SQLite база даних для конфігураційної інформації	Необхідно розробити	Доступна
<i>Обрана технологія реалізації ідеї проекту:</i> Розробляється програмний застосунок та алгоритм для розпізнавання візуальних ключів для мобільних додатків з зручним API.				

Висновок: можна зробити висновок, що технологічна реалізація проекту присутня, так як усі технології є доступними або підлягають розробці.

## 5.3 Аналіз ринкових можливостей запуску стартап-проекту

Таблиця 5.4. Попередня характеристика потенційного ринку

№	Показники стану ринку	Характеристика
1	Кількість головних гравців, од	200
2	Загальний обсяг продаж, грн./ум.од	500 ум.од в рік
3	Динаміка ринку	Зростає
4	Наявність обмежень для входу	Потребує рекламної компанії
5	Специфічні вимоги до стандартизації та сертифікації	Відсутні
6	Середня норма рентабельності в галузі або по ринку, %	50%

Висновок: враховуючи кількість головних гравців по ринку, зростаючу динаміку ринку, невелику кількість конкурентів та середню норму рентабельності можна зробити висновок, що на даний момент, ринок для входження стартап-продукту є привабливим.

Таблиця 5.5. Характеристика потенційних клієнтів стартап-проекту

№	Потреба, що формує ринок	Цільова аудиторія	Відмінності у поведінці цільових груп клієнтів	Вимоги споживачів до товару
1	Розпізнавання об'єктів на зображеннях	Розробники AR систем	Легкість інтеграції	Можливість в легкий спосіб використовувати систему як модуль в своїй програмній системі. Швидкодія та якісь розпізнавання заданих ключів.
2	Організація великих івентів	Організатори	Закінченість рішення	
3	Креативна реклама	Креативні менеджери	Легкість інтеграції	

Таблиця 5.6. Фактори загроз

№	Фактор	Зміст загрози	Можлива реакція компанії
1	Конкуренти	Наявність конкурентів котрі надають схожі рішення	Зменшення ціни на поставлену послугу; Розробка унікальних характеристик товару; Надання ліцензій на обслуговування
2	Кошти на розробку та підтримку продукту	Закінчення грошей та недостатнє фінансування	Ітеративна розробка продукту задля покрокового виведення продукту на ринок та отримання відповіді користувачів

## Закінчення таблиці 5.6

3	Вихід аналогу	Вихід аналогу даного товару може призвести до знецінення та безідейності даного товару	Вихід товару на ринок в коротші строки з не повною, але достатньою, функціональністю для зацікавлення усіх цільових аудиторій; Проведення рекламної компанії
---	---------------	--	---

Таблиця 5.7. Фактори можливостей

№	Фактор	Зміст можливості	Можлива реакція компанії
1	Новий продукт	Вихід на ринок, Зменшення монополії, Надання нових рішень у сфері	Розробка нової функціональності; Вихід нової продукції на ринок; Надання різноманітних типів ліцензій в залежності від потреб користувача \ замовника
2	Вихід аналогу	Надати продукт з певними характеристиками та можливостями що відсутні у компаній конкурентів	Аналіз ринку та користувачів задля задоволення їх потреб та надання функціональності у найкоротші строки за ціну, котра є дешевшою ніж у продуктів-замінників
3	Зворотній зв'язок від користувачів	Можливість отримання необхідної інформації для вдосконалення продукту	Наявність вхідних даних та реакція на них з боку команди розробників задля задоволення потреб та бажань кінцевих користувачів системи
4	Грошова винагорода за рекламу	При достатньому попиту на систему кешування даних можлива комерціалізація продукту на основі реклами задля отримання грошової винагороди для подальшого розвитку продукту та оплати заробітної плати працівникам	Точкова комерціалізація продукту; Введення реклами; Ведення додаткових коштів у проект задля його подальшого розвитку

Таблиця 5.9. Ступеневий аналіз конкуренції на ринку

№	Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)
1	Тип конкуренції: олігополія	Є компанії з продуктом-замінником	Пропозиція ідейно нового товару
2	За рівнем конкурентної боротьби: національний	Наявність груп клієнтів з однаковими потребами на всій території ринку	Робота на якомога ширшій території
3	Галузева ознака: внутрішньогалузева	Даний тип продукту може використовуватися тільки у сфері розробки ІТ додатків \ продуктів	Надання зручного, інтуїтивно зрозумілого інтерфейсу; Підтримка всім відомих методів взаємодії з середовищем розробки; Наявність документації та он-лайн підтримки
4	Конкуренція за видами товарів: товарно-видова	Дана конкуренція – конкуренція між товарами одного виду.	Впровадження функціональності яка відсутня у товарів-замінників; Спрощення інтерфейсів; Надання підтримки
5	За характером конкурентних переваг: нецінова	Конкуренція за якісними характеристиками	Запровадження нової технології
6	За інтенсивністю: Марочна	Наявність унікального знаку що відрізняє даний продукт від продуктів-замінників	Впровадження власної назви та власного знаку

Таблиця 5.10. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	AlwaysAI AWSImage Cogniac	Детальний аналіз ринку показує, що потенційні бар'єри для виходу на ринок відсутні	Відсутні	Клієнти можуть бути не задоволені складністю інтеграції системи	Є компанії, що надають товари - замінники
Висновки	Відсутня монополія зі сторін даних конкурентів, тому вихід на ринок не має додаткових ускладнень.	Є можливість виходу на ринок. Строк виходу на ринок близько 15 місяців. Потенційні конкуренти є, проте не вдовольняють потреби ринку;	Постачальники не диктують умови роботи	Клієнти вносять побажання, що доцільно брати до уваги при розробці продукту.	Відсутні

Проаналізувавши можливості роботи на ринку з огляду на конкурентну ситуацію можна зробити висновок: оскільки кожний з існуючих продуктів не впливає у великій мірі на поточну ситуацію на ринку в цілому, кожний з існуючих продуктів має свою специфічну сферу використання та свої позитивні та негативні сторони щодо рішення певних типів задач, то робота та вихід на даний ринок є можливою і реалізованою задачею [20].

Для виходу на ринок продукт повинен мати функціонал що відсутній у продуктів-аналогів, повинен задовольняти потреби користувачів, мати необхідний та достатній функціонал з конфігурування, підтримку зі сторони розробників та можливість розробки спеціального функціоналу за відповідною ліцензією.



Таблиця 5.13. SWOT аналіз стартап-проекту

<p>Сильні сторони (S):</p> <ul style="list-style-type: none"> <li>– Якісне розпізнавання ключів</li> <li>– Легка інтеграція</li> </ul>	<p>Слабкі сторони (W):</p> <ul style="list-style-type: none"> <li>– Цінова політика</li> <li>– Складність розробки</li> </ul>
<p>Можливості (O):</p> <ul style="list-style-type: none"> <li>– Розширення сфер застосування</li> <li>– Зростання попиту на продукцію.</li> <li>– Вихід на нові ринки або сегменти.</li> <li>– Послаблення позицій конкурентів.</li> <li>– Використання новітніх світових технологій.</li> </ul>	<p>Загрози (T):</p> <ul style="list-style-type: none"> <li>– Недостатнє фінансування</li> <li>– Поява конкурентів</li> </ul>

Таблиця 5.14. Альтернативи ринкового впровадження стартап-проекту

№	Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
1	Безкоштовне надання певного функціоналу у користування споживачам на обмежений термін	Головний ресурс – люди, даний ресурс – наявний	2-3 місяці
2	Реклама	Залучення власних коштів для реклами товару	1-2 місяці
3	Написання статей та опис товару на відомих ресурсах	Головний ресурс – час, даний ресурс – наявний	2-3 тижні
4	Презентація товару на хакатонах й інших ІТ заходах	Ресурс – час та гроші для участі, наявні	1-3 місяці



## 5.4 Розроблення ринкової стратегії проекту

Таблиця 5.15. Вибір цільових груп потенційних споживачів

№	Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтовний попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу у сегмент
1	Розробники AR систем	Висока зацікавленість	Попит середній	Висока	Низька
2	Креативні менеджери	Середня зацікавленість	Попит середній	Середня	Висока
Які цільові групи обрано: краще всього підійде група 1, оскільки простота входу і зацікавленість прийнятні					

Відповідно до проведеного аналізу можна зробити висновок, що підходящою цільовою групою для розповсюдження даного програмного продукту є працівники ІТ сфери, ІТ компанії в цілому та будь-які підприємства котрі використовують програмні продукти побудовані на будь-якій мові програмування

та мають потребу в функціоналі верифікації за зображенням. Відповідно до стратегії охоплення ринку збуту товару обрано стратегію масового маркетингу, оскільки для підприємств, ІТ працівників та ІТ компаній у цілому надається стандартизований продукт з можливістю розширення функціональності за домовленістю (відповідно до ліцензії).

Таблиця 5.16. Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Надання функціональності що відсутня у товарів-замінників, підтримка клієнтів	Проведення реклами, освітлення унікальної функціональності через інтернет ресурси та інші канали, контакт напряду з споживачами; формування лояльності і прихильності споживачів	Зниження ступеню замінності товару; Прихильність клієнтів; Відмітні властивості товару; Відмітні характеристики товару;	Стратегія диференціації

Таблиця 5.17. Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, які?	Стратегія конкурентної поведінки
Ні, оскільки є товари-замінники, але дані товари замінники не мають деякого необхідного функціоналу	Так, ціль компанії знайти нових споживачів та, частково, забрати існуючих у конкурентів задля задоволення потреб останніх	Компанія частково копіює характеристики товару конкурента, основна ціль компанії розробка нового унікального функціоналу, з підтримкою основного функціоналу конкурентів	Стратегія заняття конкурентної ніші

Таблиця 5.18. Визначення стратегії позиціонування

№	Вимоги до товару цільової аудиторії	Базова стратегія розвитку	Ключові конкурентоспроможні позиції власного стартап-проекту	Вибір асоціацій, які мають сформувати комплексну позицію власного проекту
1	Зручність в інтеграції	Стратегія диференціації	Зрозумілий та якісний API інтерфейс	Легкість в інтеграції
2	Швидкодія	Стратегія диференціації	Якісна нейронна система розпізнавання ключів	Якісна і швидка верифікація ключів

Відповідно до проведеного аналізу можна зробити висновок, що стартап-компанія вибирає як базову стратегію розвитку – стратегію диференціації, як базову стратегію конкурентної поведінки – стратегію заняття конкурентної ніші.

### 5.5 Розроблення маркетингової програми стартап-проекту

Таблиця 5.19. Визначення ключових переваг концепції потенційного товару

№	Потреба	Вигода, яку пропонує товар	Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити)
1	Швидкодія	Швидке розпізнавання зображень	Швидше розпізнавання ключів
2	Легкість в інтеграції	Легка інтеграція	Легка інтеграція в існуючі системи
3	Широка підтримка	Допомога спеціалістів	Допомога спеціалістів включена в ціну
4	Вибір ключів	Вибір ключа для верифікації	Можливість вибрати вже готовий ключ

Таблиця 5.20. Опис трьох рівнів моделі товару

Рівні товару	Сутність та складові		
1. Товар за задумом	Система верифікування візуальних ключів для мобільних додатків		
2. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх/Тл/Е/Ор
	Легкість інтеграції	-	Висока
	Швидкодія	-	Висока
	Вартість одного розпізнавання	грн	2
	Вартість повного пакету	грн	
	Якість: пройшла ручне тестування, та бета тестування		
	Доступна за адресою в мережі Інтернет		
Марка: VerifyNet, назва: Розпізнавання візуальних ключів			
3. Товар із підкріпленням	До продажу: наявна повна документація, акції на придбання декількох ліцензій, знижки для певних сегментів на покупку товару		
	Після продажу: додаткова підтримка спеціалістів налаштування, підтримка з боку розробника		
Проект буде захищено від копіювання реєстрацією назви програми, створення заявки на отримання патенту на винахід, щоб уберегти алгоритм роботи від копіювання			

Таблиця 5.21. Визначення меж встановлення ціни

Рівень цін на товари-замінники	Рівень цін на товари-аналоги	Рівень доходів цільової групи споживачів	Верхня та нижня межі встановлення ціни на товар/послугу
Замінники відсутні	1 цент за верифікацію	Вище середнього	1 цент за одне зображення

Таблиця 5.22. Формування системи збуту

Специфіка закупівельної поведінки цільових клієнтів	Функції збуту, які має виконувати постачальник товару	Глибина каналу збуту	Оптимальна система збуту
Оплата підписок	Електронний вигляд	Виробник-споживач	Сайт виробника

Таблиця 5.23. Концепція маркетингових комунікацій

№	Специфіка поведінки цільових клієнтів	Канали комунікацій, якими користуються цільові клієнти	Ключові позиції, обрані для позиціонування	Завдання рекламного повідомлення	Концепція рекламного звернення
1	немає	Інтернет	Якість послуги	Поширення інформації про рішення, його функціонал та переваги перед конкурентами	Звернення засноване на виділені якісних характеристик розробки

Як результат було створено ринкову (маркетингову) програму, що включає в себе визначення ключових переваг концепції потенційного товару, опис моделі товару, визначення меж встановлення ціни, формування системи збуту та концепцію маркетингових комунікацій.

#### Висновки по розділу

В п'ятому розділі описано стратегії та підходи з розроблення стартап-проекту, визначено наявність попиту, динаміку та рентабельність роботи ринку, як висновок було вказано що існує можливість ринкової комерціалізації проекту. Розглянувши потенційні групи клієнтів, бар'єри входження, стан конкуренції та конкурентоспроможність проекту було встановлено що проект є перспективним. Розглянуто та вибрано альтернативу впровадження стартап-проекту та доведено доцільність подальшої імплементації проекту.

## ВИСНОВКИ

Під час виконання цієї магістерської дисертації, було проаналізовано проблеми розпізнавання зображень та знаходження об'єктів на них. Також розглянуто існуючі рішення цих проблем та існуючі аналоги програмних застосунків, що вирішують ці проблеми.

Проаналізовано існуючі методології знаходження об'єктів на зображеннях та їх класифікацію. Обрано найбільш підходящу методику яка передбачає використання згорткових нейронних мереж.

Після сформування завдання, було проаналізовано та обрано для реалізації програмного продукту мову програмування, фреймворки, алгоритмічні рішення, середовище для розгортки програмного продукту.

Після чого було проведено дослідження роботи нейронної мережі. Дослідження показало ефективність роботи алгоритму.

Для зручного доступу спроектований дизайн API з зручною та зрозумілою документацією, що надає можливість використовувати систему в мобільних додатках з мінімальними затратами часу на інтеграцію.

Було створено модель роботи даного програмного застосунку як стартап проекту.

Тепер дана система дозволяє верифікувати візуальні ключі на зображеннях, що робить її зручним інструментом в розробці додатків найрізніших типів.

## ПЕРЕЛІК ПОСИЛАНЬ

1. Imagenet large scale visual recognition challenge 2016. URL: [http:// image-net.org/challenges/LSVRC/2016](http://image-net.org/challenges/LSVRC/2016) (дата звернення 2017-04-07).
2. Infotrends – how long does it take to shoot 1 trillion photos?. URL: <http://blog.infotrends.com/?p=21573> (дата звернення 2019-06-17).
3. Matconvnet: Cnns for matlab. URL: <http://www.vlfeat.org/matconvnet/> (дата звернення 2019-01-10).
4. Software at the personal website of derek hoiem. URL: <http://dhoiem.cs.illinois.edu/software/> (дата звернення 2018-02-17).
5. Зайцев И.В. Нейронные сети: основные модели. Учебное пособие для физ. ф-та. Воронеж. 1991.
6. Bishop, C. M. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
7. Головки В.А. Нейронные сети: обучения, организация и применение. М.: ИПРЖР. 2008
8. S. Singh, “Critical reasons for crashes investigated in the national motor vehicle crash causation survey,” 2015
9. Liu, C.; Chang, F.; Chen, Z.; Liu, D. Fast traffic sign recognition via high-contrast region extraction and extended sparse representation. IEEE Trans. Intell. Transp. Syst. 2016, 17, 79–92
10. Varun, S.; Singh, S.; Kunte, R.S.; Samuel, R.S.; Philip, B. An object recognition system based on template matching employing tree classifier. In Proceedings of the International Conference on Computational Intelligence and Multimedia Applications ICCIMA, Sivakasi, India, 13–15 December 2007; pp. 360–365.
11. Ruta, A.; Li, Y.; Liu, X. Detection, tracking and recognition of image objects from video input. In Proceedings of the 11th International IEEE Conference on

- Intelligent Transportation Systems, ITSC 2008, Beijing, China, 12–15 October 2008; pp. 55–60.
12. Jiang, Y.; Zhou, S.; Jiang, Y.; Gong, J.; Xiong, G.; Chen, H. Objects recognition using ridge regression and Otsu method. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden-Baden, Germany, 5–9 June 2011; pp. 613–618.
  13. Vázquez-Reina, A.; Lafuente-Arroyo, S.; Siegmann, P.; Maldonado-Bascón, S.; Acevedo-Rodríguez, F. Objects shape classification based on correlation techniques. In Proceedings of the 5th WSEAS International Conference on Signal Processing, Computational Geometry & Artificial Vision, Alcalá de Henares, Malta, 15–17 September 2005; pp. 149–154.
  14. Maldonado-Bascón, S.; Lafuente-Arroyo, S.; Gil-Jimenez, P.; Gómez-Moreno, H.; López-Ferreras, F. Image detection and recognition based on support vector machines. *IEEE Trans. Intell. Transp. Syst.* 2007, 8, 264–278.
  15. Jiménez, P.G.; Bascón, S.M.; Moreno, H.G.; Arroyo, S.L.; Ferreras, F.L. Image sign shape classification and localization based on the normalized FFT of the signature of blobs and 2D homographies. *Signal Process.* 2008, 88, 2943–2955.
  16. Hoiem D., Efros A., Efros A., and Hebert, M. Putting objects in perspective. *International Journal of Computer Vision* 80. 2008. P. 3–35.
  17. Nicchiotti, G.; Ottaviani, E.; Castello, P.; Piccioli, G. Classification from color image sequences. In Proceedings of the 7th International Conference on Image Analysis and Processing, 1994; pp. 623–626.
  18. Priebe, L.; Rehrmann, V. On hierarchical color segmentation and applications. In Proceedings of the 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, NY, USA, 15–17 June 1993; pp. 633–634.
  19. Swain, M.J.; Ballard, D.H. Color indexing. *Int. J. Comput. Vis.* 1991, 7, 11–32.



20. Fan, J.; Yau, D.K.; Elmagarmid, A.K.; Aref, W.G. Automatic image segmentation by integrating color-edge extraction and seeded region growing. *IEEE Trans. Image Process.* 2001, 10, 1454–1466.
21. Розроблення стартап-проекту [Електронний ресурс] : Методичні рекомендації до виконання розділу магістерських дисертацій для студентів інженерних спеціальностей / За заг. ред. О.А. Гавриша. – Київ : НТУУ «КПІ», 2016. – 28 с.
22. Олексій А.В., Корнага Я.І. Інтелектуальна система розпізнавання візуальних ключів для мобільних додатків. *Вчені записки Таврійського Національного Університету Імені В.І. Вернадського. Том 30(69) №6 2019 частина 2.* С. 31-35.

## ДОДАТКИ

## **ДОДАТОК А ГРАФІЧНИЙ МАТЕРІАЛ**

## ПЛАКАТ 1 Структура мережі Fast R-CNN VGG-16

## ПЛАКАТ 2 Схеми роботи методу CheckVisualKey

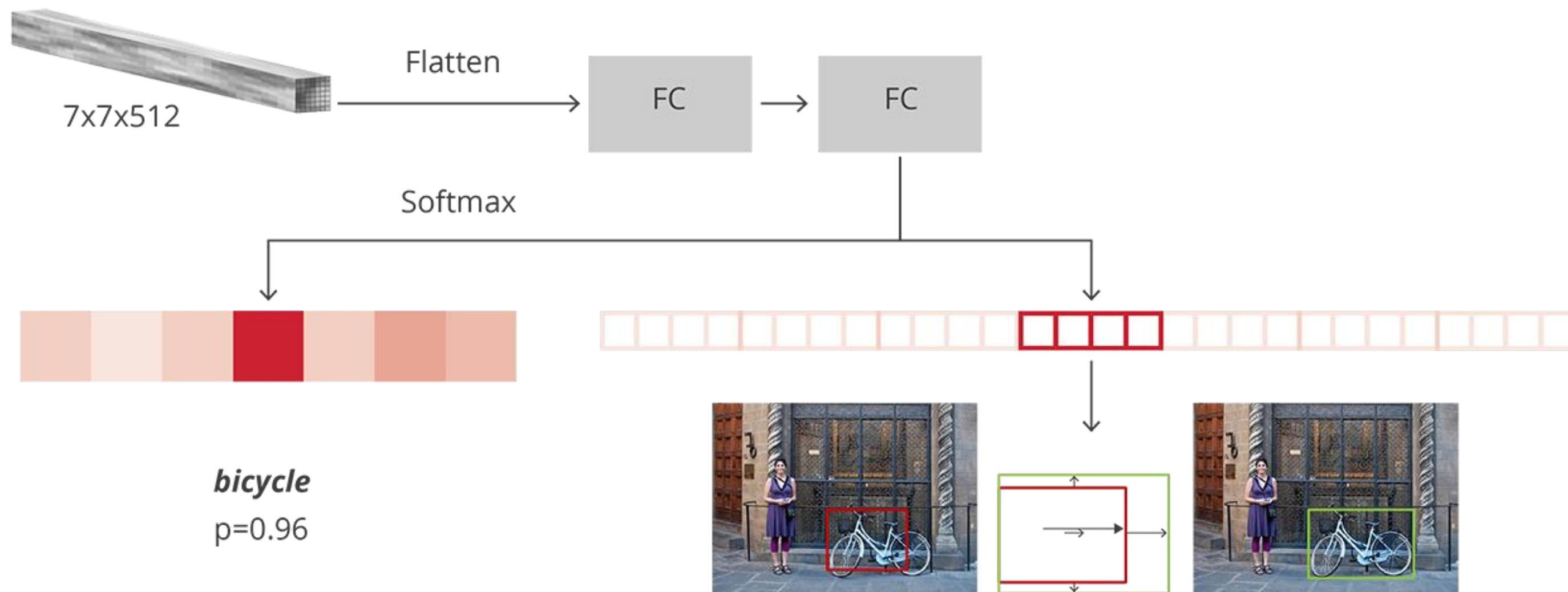
## ПЛАКАТ 3 Результати експериментів

## ПЛАКАТИ 4 Приклад розпізнавання тестових даних в різних конфігураціях

**ДОДАТОК Б РЕЗУЛЬТАТ ПРОХОДЖЕННЯ ПЕРЕВІРКИ НА ПЛАГІАТ**



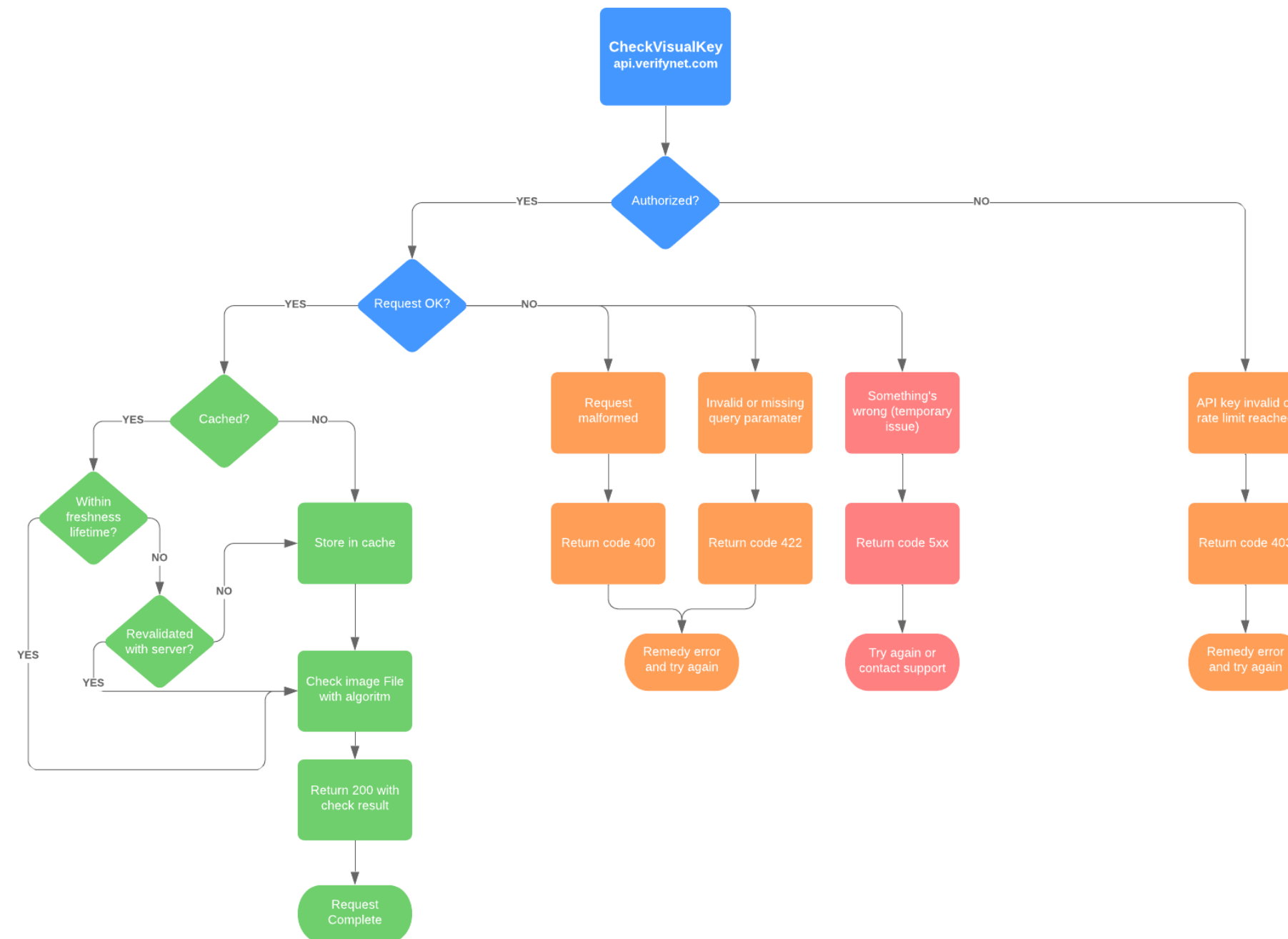
# Структура мережі Fast R-CNN VGG-16



Демонстраційний плакат №1  
до магістерської дисертації на тему  
„Інтелектуальна система розпізнавання візуальних ключів для  
мобільних додатків”

Розробив: Олексій А.В.  
Прийняв: Корнага Я.І.

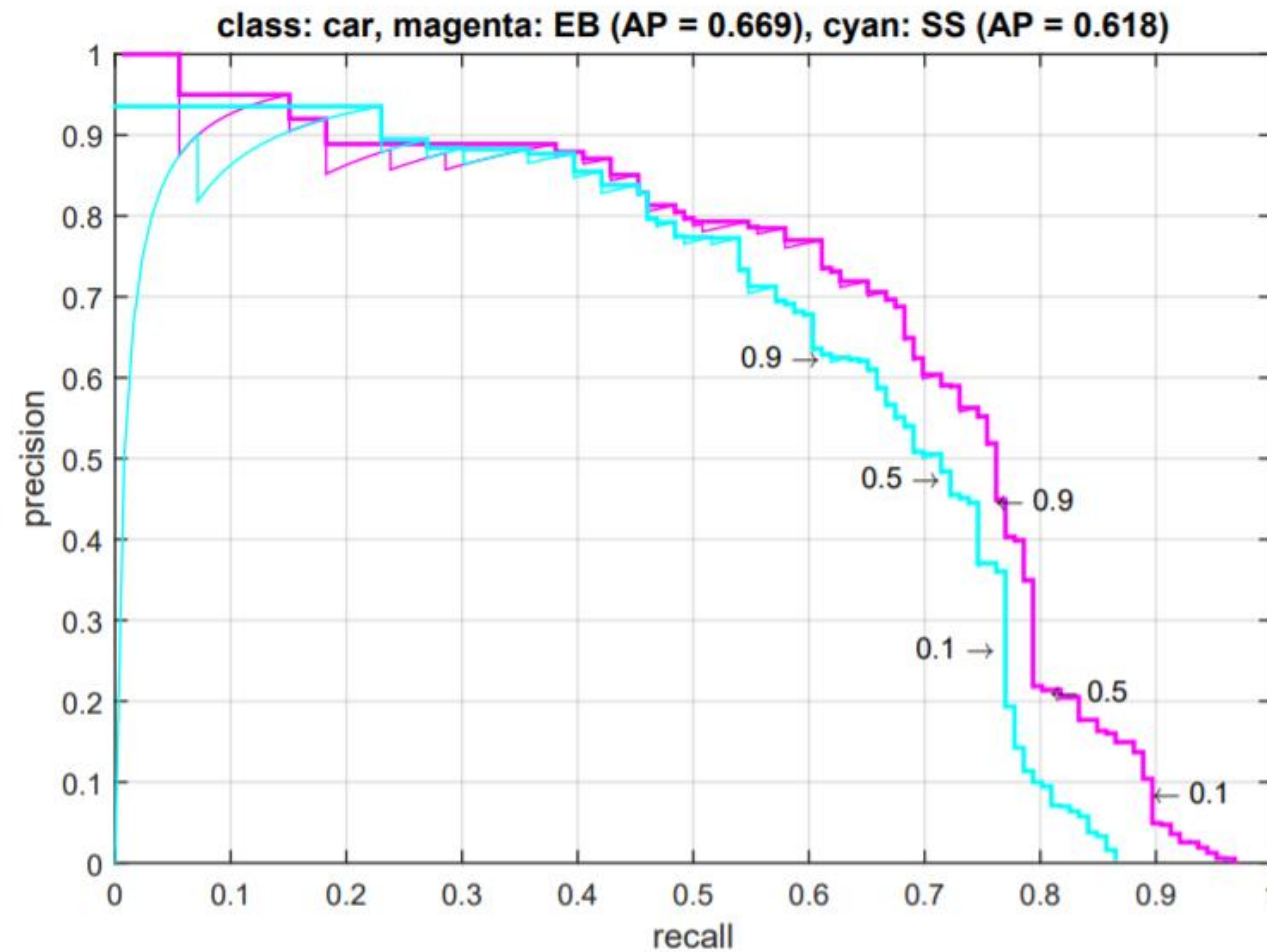
# Схема роботи методу CheckVisualKey



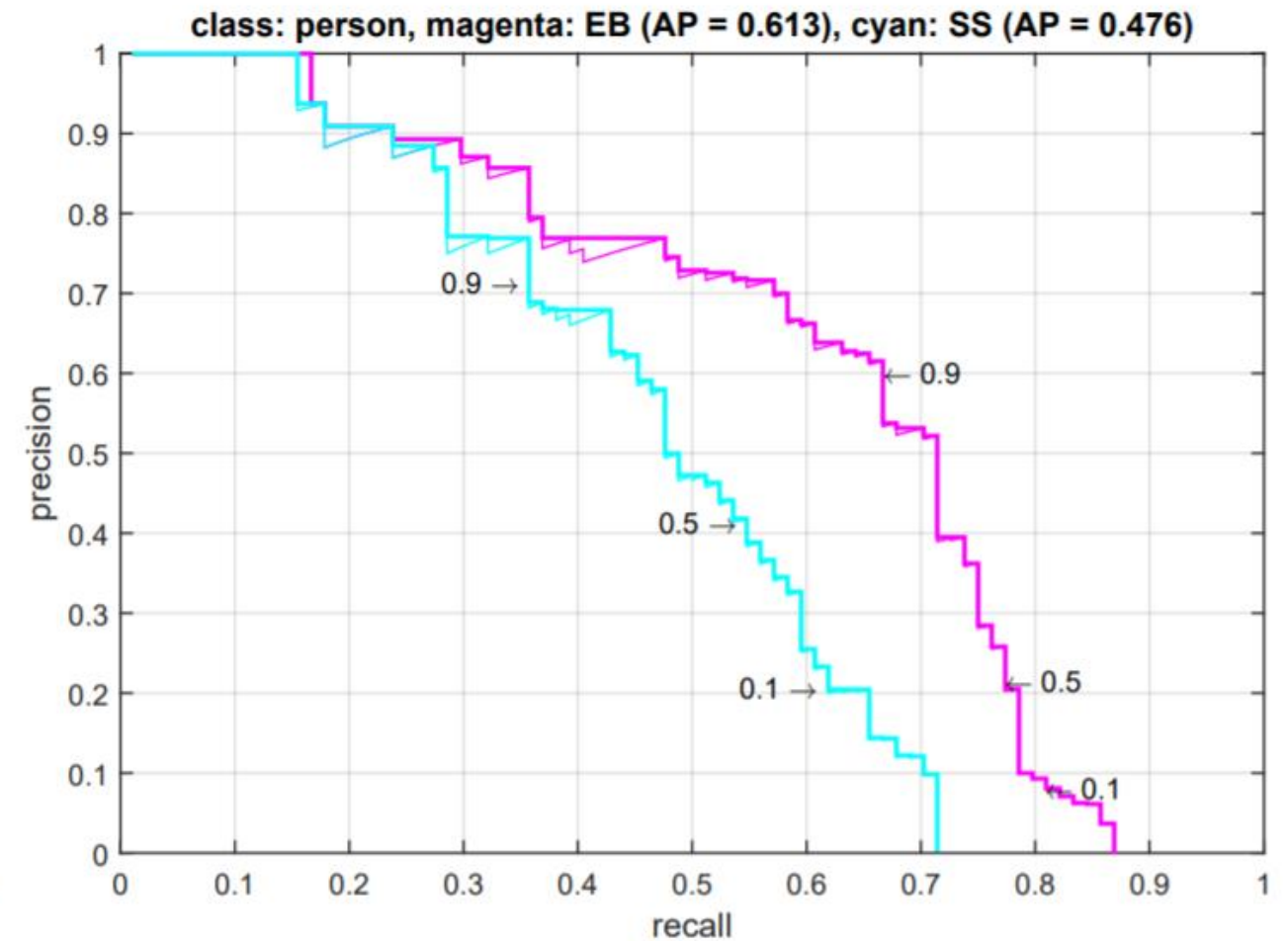
Демонстраційний плакат №2  
до магістерської дисертації на тему  
„Інтелектуальна система розпізнавання візуальних ключів для  
мобільних додатків”

Розробив: Олексій А.В.  
Прийняв: Корнага Я.І.

# Результати експериментів



(a) Car class



(b) Pedestrian class

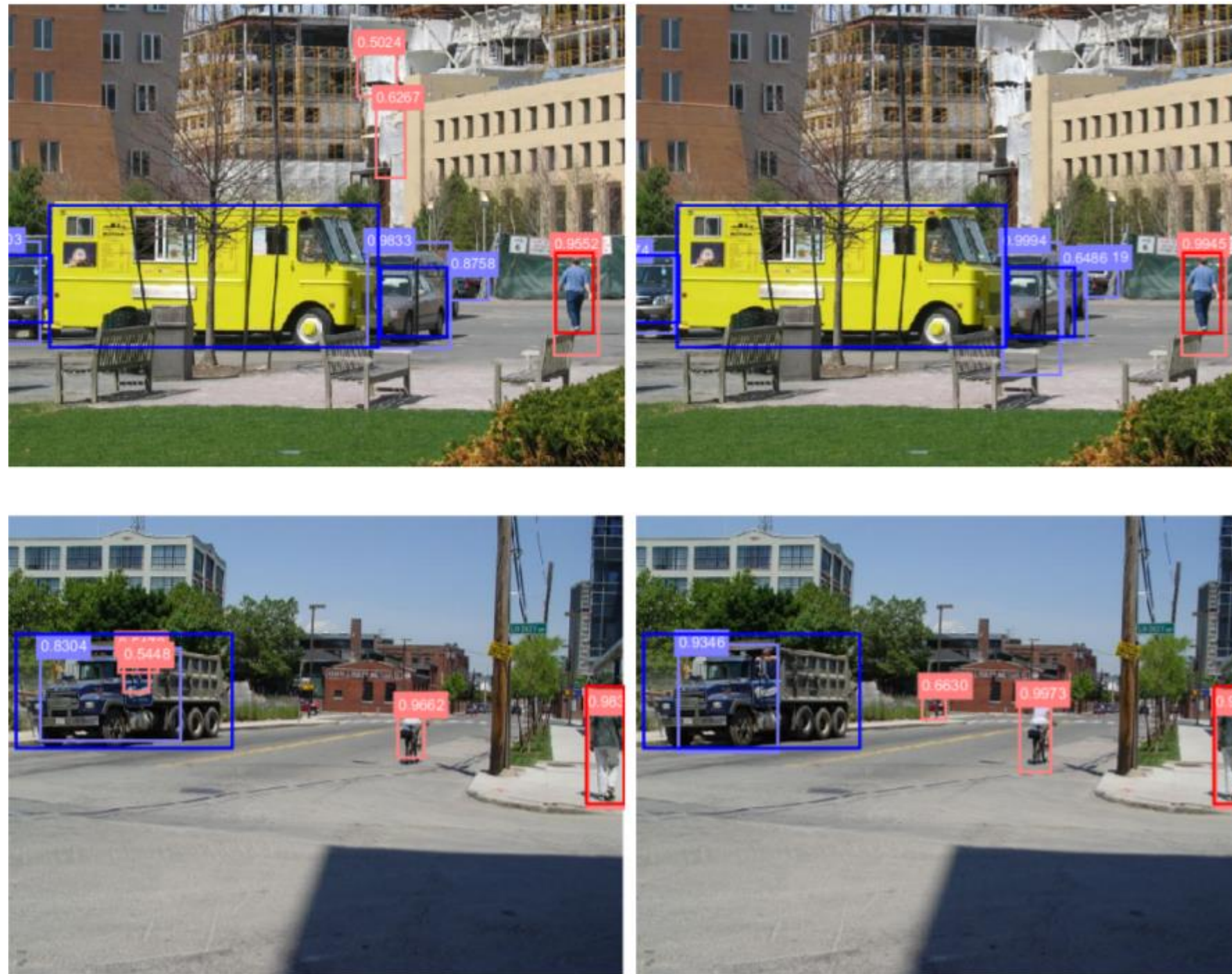
Демонстраційний плакат №3  
до магістерської дисертації на тему  
„Інтелектуальна система розпізнавання візуальних ключів для  
мобільних додатків”

Розробив: Олексій А.В.

Прийняв: Корнага Я.І.



# Приклад розпізнавання тестових даних в різних конфігураціях



Демонстраційний плакат №4  
до магістерської дисертації на тему  
„Інтелектуальна система розпізнавання візуальних ключів для  
мобільних додатків”

Розробив: Олексій А.В.  
Прийняв: Корнага Я.І.